A Self-consistent Model for Polycrystal Deformation

Description and Implementation

Bjørn Clausen and Torben Lorentzen

Abstract   This report is a manual for the ANSI C implementation of an incremental elastic-plastic rate-insensitive self-consistent polycrystal deformation model based on (Hutchinson 1970). The model is furthermore described in the Ph.D. thesis by Clausen (Clausen 1997). The structure of the main program, `sc_model.c`, and its subroutines are described with flow-charts. Likewise the pre-processor, `sc_ini.c`, is described with a flowchart. Default values of all the input parameters are given in the pre-processor, but the user is able to select from other pre-defined values or enter new values. A sample calculation is made and the results are presented as plots and examples of the output files are shown.

Contents

1 Introduction

The modelling of plastic deformation of polycrystalline solids have been addressed within the field of materials science for the last 7 decades or so since the early modelling efforts of Sachs (Sachs 1928) and Taylor (Taylor 1938). Today the state-of-the-art for the so-called 1-site models are various implementations of self-consistent modelling schemes, where the interaction between grains in the polycrystalline aggregate is taken into account in more or less realistic manners.

The present report describes in detail the implementation of a specific self-consistent modelling scheme introduced by (Hutchinson 1970). This implementation forms the basis for a detailed description of selected numerical results, which is found in the Ph.D thesis by Clausen (Clausen 1997).

Following a short description of the mathematical modelling approach in section 2, the implementation is described in section 3 through a series of successive flow-charts with added comments specifying the parameters to be evaluated in each step of the sequential execution. Appendix A gives a complete list of all subroutines of the code, and Appendix B describes relevant aspects of the nomenclature used.

Section 4 contains a sample calculation presenting selected numerical results with specific emphasis on quantities related to the crystallographic slip pattern and on quantities like elastic strains in selected sub-sets of grains with specific orientations. The first group of results are of a fundamental interest within the field of modelling the plastic deformation of crystalline solids while the latter group of results are of great relevance for the utilisation of diffraction techniques for stress/strain characterisation both in materials science and in engineering.

The numerical code, which is written in ANSI C, is available at Risø by contacting Senior Scientist, Dr. T. Lorentzen (e-mail: torben.lorentzen@@risoe.dk). The code is portable to any platform containing an ANSI C compiler, and the sample calculation provides a base for users to verify the execution of the code on their own platform. Appendix C gives examples on the format of the output files from the execution. Section 5 gives a brief description of the software and hardware requirements for an execution of the code.

2 General model description

The present implementation is based on the self-consistent modelling scheme proposed in (Hutchinson 1970). It is an incremental elastic-plastic rate-insensitive self-consistent polycrystal deformation model. The *fcc* polycrystal is regarded as an agglomerate of single crystals and thereby the model is governed by the single crystal slip mechanisms with the controlling parameters being the critical resolved shear stress and the hardening law. In the self-consistent scheme the grains are considered as spherical inclusions in an infinite continuum matrix, where the properties of the matrix is determined as the average of all the grains. This means that the elastic and plastic interaction between the grain and the continuum matrix are taken into account, but the direct grain-to-grain interaction, as in real materials, are not. The model is a small strain model as the strain definition is without second order terms and it does not include localisation or instabilities such as necking. Only *fcc* polycrystals are considered, but the model can easily be extended to *bcc* and *hcp* if the allowable slip systems and the hardening parameters are provided.

The initial critical resolved shear stress, τ_0, is assumed to be the same on all 12 well known $\{110\}\langle 111\rangle$ slip systems in the *fcc* single crystals. The plastic strain rate in the constituents, $\dot{\varepsilon}_c^p$, is the sum of the contributions of the shear rates from all the active slip systems

$$\dot{\varepsilon}_c^p = \sum_i \dot{\gamma}^i \boldsymbol{\mu}^i \tag{2.1}$$

where

$$\mu_{kl}^i = \frac{1}{2}\left(m_k^i n_l^i + m_l^i n_k^i\right) \quad \text{(no sum on } i) \tag{2.2}$$

and $\boldsymbol{n}$ and $\boldsymbol{m}$ are slip plane normal and slip direction vectors. The total strain rate is the sum of the elastic and plastic part as given by

$$\dot{\varepsilon}_c = \boldsymbol{\mathcal{M}}_c \dot{\boldsymbol{\sigma}}_c + \dot{\varepsilon}_c^p \quad \text{or} \quad \dot{\boldsymbol{\sigma}}_c = \boldsymbol{\mathcal{L}}_c \left(\dot{\varepsilon}_c - \dot{\varepsilon}_c^p\right) \tag{2.3}$$

where $\boldsymbol{\mathcal{L}}_c$ and $\boldsymbol{\mathcal{M}}_c$ are the elastic moduli and compliance for the single crystals.

The current critical resolved shear stress of the *i*th slip system is denoted τ^i and the rate of the critical resolved shear stress is found from the hardening law. The hardening properties for multi slip in a single crystal are difficult to determine and the assumptions used in most polycrystal deformation models are purely empiric. Normally the critical resolved shear stress rates are assumed to be related linearly to the shear rates (Hill 1966). The distinguishing between self and latent hardening are incorporated as shown in equation 2.4;

$$\dot{\tau}^i = \sum_j h^{ij} \dot{\gamma}^j \quad \text{where} \quad h^{ij} = h_\gamma \left(q + (1-q)\delta^{ij}\right) \tag{2.4}$$

where δ^{ij} is Kronecker's delta. The factor q determines the degree of latent hardening, *e.g.* $q = 0$ provides only self hardening, $q = 1$ provides Taylor hardening and $q > 1$ provides stronger latent hardening than self hardening.

The hardening in the grains are dependent on previous deformation history. The normal assumption is to make the instantaneous hardening coefficient for a grain, h_γ, dependent on the accumulated slip in the grain, γ^{acc}. Apart from the obvious linear first approximation, various functions have been used to describe this relationship. In the present work we are interested in a good agreement between the macroscopic stress-strain response of the model and a measured work hardening curve for the materials. In the present implementation the relationship between γ^{acc} and h_γ is described with an exponentially decreasing function;

$$h_\gamma = h_{final}\left(1 + (h_{ratio} - 1)\,e^{(-h_{exp}\gamma^{acc})}\right) \tag{2.5}$$

where h_{final} is the final hardening coefficient, h_{ratio} is the ratio between the initial and the final hardening coefficient and h_{exp} is a parameter that determines the strength of the exponential part. This formulation then includes the simple linear hardening law that is obtained if $h_{ratio} = 1$ and equation 2.5 reduces to $h_\gamma = h_{final}\gamma^{acc}$. The three empiric parameters h_{final}, h_{ratio} and h_{exp} have no physical foundation and are used as fitting parameters to make the macroscopic response of the model resemble the real material behaviour. Selection of τ_0 and the hardening coefficients, are not trivial. As a rule, τ_0 is assumed equal to half of the largest principal stress difference at yield, where yield is defined as the point where the first grain becomes plastic; a point that is difficult to determine from a stress-strain curve. In the present model, τ_0 is selected together with the hardening coefficients to make the predictions of the macroscopic stress-strain curves resemble the actual material behaviour. In practise it is necessary to determine τ_0 and the hardening coefficients by iteration. Taylor hardening ($q = 1$) must be avoided and h_{final} must be non-zero as these conditions cause numerical problems in the slip rates calculation where the set of equations to be solved are no longer independent. In practice q values that differ more than one per cent from 1 and h_{final} values larger than 1 are applicable.

The active slip systems have the resolved shear stress equal to τ and the increment in the resolved shear stress equal to $\dot{\tau}$

$$\boldsymbol{\sigma}_c \boldsymbol{\mu}^i = \tau^i \quad \text{and} \quad \dot{\boldsymbol{\sigma}}_c \boldsymbol{\mu}^i = \dot{\tau}^i \tag{2.6}$$

The elastic-plastic instantaneous stiffness tensor for the grain, $\boldsymbol{L}_c$, is determined by combining equations 2.1, 2.3, 2.4 and 2.6 for the N active slip systems

$$\sum_j \dot{\gamma}^j X^{ij} = \boldsymbol{\mu}^i \boldsymbol{\mathcal{L}}_c \dot{\boldsymbol{\varepsilon}}_c , \qquad X^{ij} = h^{ij} + \boldsymbol{\mu}^i \boldsymbol{\mathcal{L}}_c \boldsymbol{\mu}^j \tag{2.7}$$

where X^{ij} and its inverse (Y^{ij}) are $N \times N$ matrices. The slip on the active systems is found as

$$\dot{\gamma}^i = \boldsymbol{f}^i \dot{\boldsymbol{\varepsilon}} , \qquad \boldsymbol{f}^i = \sum_k Y^{ik} \boldsymbol{\mathcal{L}}_c \boldsymbol{\mu}^k \tag{2.8}$$

where $\boldsymbol{f}^i$ is an N vector. $\boldsymbol{L}_c$ is then found as

$$\boldsymbol{L}_c = \boldsymbol{\mathcal{L}}_c \left(\boldsymbol{I} - \sum_m \boldsymbol{\mu}^m \boldsymbol{f}^m \right) \tag{2.9}$$

where the sum is formed by the uncontracted products $\mu_{ij}^m f_{kl}^m$. If all slip systems are inactive the instantaneous stiffness tensor reduces to the elastic stiffness tensor, $\boldsymbol{L}_c = \boldsymbol{\mathcal{L}}_c$.

The calculations of the constituent stress and strain rates are done via fourth order concentration tensors in the way proposed by Hill (Hill 1965a, Hill 1965b). The grains are approximated by spherical single crystals embedded in an infinite homogeneous matrix whose moduli are the overall polycrystal instantaneous moduli to be determined. In this way the interaction between the grain under consideration and the matrix/polycrystal is taken into account, but the direct grain-to-grain interaction is not incorporated in the model. The stress and strain rates in the spherical inclusions are uniform (Eshelby 1957) and related to the stress and strain rates at infinity by fourth order concentration tensors, $\boldsymbol{A}_c$, according to

$$\dot{\boldsymbol{\varepsilon}}_c = \boldsymbol{A}_c \dot{\bar{\boldsymbol{\varepsilon}}} , \qquad \boldsymbol{A}_c = (\boldsymbol{L}^* + \boldsymbol{L}_c)^{-1} (\boldsymbol{L}^* + \boldsymbol{L}) \tag{2.10}$$

where $\boldsymbol{L}^*$ is Hill's 'constraint' tensor that relates to the Eshelby tensor $\boldsymbol{S}$ as

$$\boldsymbol{L}^* \boldsymbol{S} = \boldsymbol{L}(\boldsymbol{I} - \boldsymbol{S}) \tag{2.11}$$

The relationship between the Eshelby tensor, S, and the stiffness tensor, L, is given by the fourth order tensor Λ (Kneer 1965) and using equation 2.11 the constraint tensor can be found as

$$L^* = \Lambda^{-1} - L \tag{2.12}$$

Λ is determined from the double integral

$$\Lambda_{ijmn} = \frac{1}{16\pi} \int\limits_{\theta=0}^{\pi} \int\limits_{\phi=0}^{2\pi} \left(\hat{U}_{im} k_n k_j + \hat{U}_{jm} k_n k_i + \hat{U}_{in} k_m k_j + \hat{U}_{jn} k_m k_i \right) \sin\theta \mathrm{d}\theta \mathrm{d}\phi$$

$$\tag{2.13}$$

where the second order tensor $\hat{U}$ is found from

$$L_{ijkl} \hat{U}_{km} k_j k_l = \delta_{im} \tag{2.14}$$

and $k_1 = \sin\theta\cos\phi$, $k_2 = \sin\theta\sin\phi$ and $k_3 = \cos\theta$.

The stress and strain rates at infinity are identified with the polycrystal quantities $\dot{\bar{\sigma}}$ and $\dot{\bar{\varepsilon}}$. Thus, the constituent stress and strain rates are estimated by the solution to a problem in linear anisotropic elasticity.

The polycrystal stress and strain rates are equal to the weighted stress and strain rates average of all the grains, and if the average of all the grains is denoted by { }, then

$$\{\dot{\sigma}_c\} = \dot{\bar{\sigma}} \;\Rightarrow\; L = \{L_c A_c\} \tag{2.15}$$

At a certain stage of deformation the stress and thereby the potentially active slip systems in the constituents of the polycrystal are known. The polycrystal is prescribed an additionally strain rate, $\dot{\bar{\varepsilon}}$, and using the present model, we are able to determine the stress and strain rates, $\dot{\sigma}_c$ and $\dot{\varepsilon}_c$, as well as the instantaneous single crystal moduli, L_c, for all the grains and the polycrystalline quantities, $\dot{\bar{\sigma}}$ and L, are determined as the appropriate average of all the grains.

3 Implementation

The implementation of the described model is divided into two programs; the pre-processor (`sc_ini.c`) and the main program (`sc_model.c`) as shown in the flowchart in figure 3.1

```
   ┌──────────────┐
  (    Start      )
   └──────┬───────┘
   ┌──────┴───────┐
   │ Pre-processor│      Interactive execution rendering an ASCII input file
   │  sc_ini.c    │      (sc_input.dat) for the main program.
   └──────┬───────┘
   ┌──────┴───────┐
   │ Main Program │      The main program is calculating the material be-
   │  sc_model.c  │      haviour using the present polycrystal model.
   └──────┬───────┘
   ┌──────┴───────┐
  (    return     )
   └──────────────┘
```
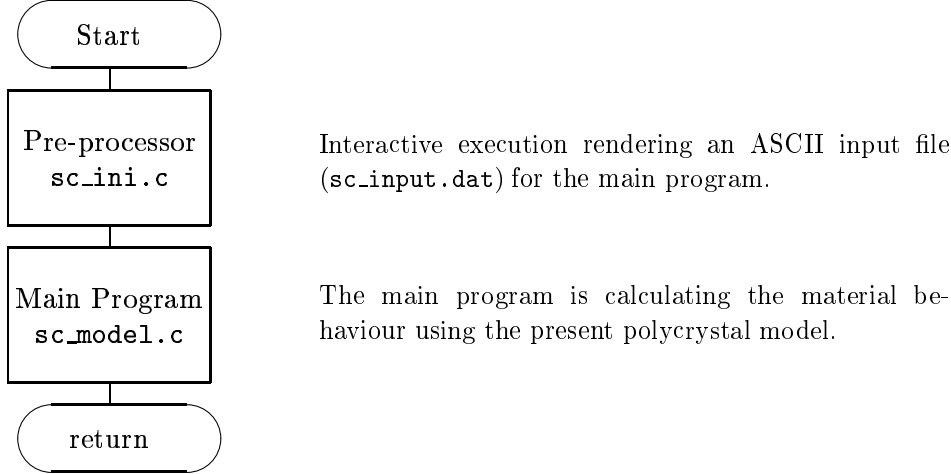
Figure 3.1. Flow chart for the program.

When running the pre-processor default values to all the needed information are suggested though the user is able to type other input or choose from preselected items. The result is an input file (`sc_input.dat`) for the main program. The main program proceeds to calculate the overall stress-strain response of the polycrystal as well as a range of the elastic strains in selected subsets of grains representing the Bragg reflections in a neutron diffraction experiment.

The subroutines in `sc_model.c` and the auxiliary subroutines from 'Numerical recipes in C' (Press *et al.* 1992), placed in the file `sc_aux.c`, are listed in table A.1 in appendix A.

3.1 The pre-processor (`sc_ini.c`)

The following is a description of the sequential interactive execution of the pre-processor. The flow chart is shown in figure 3.2 while details are given below.

The pre-processor produces an ASCII input file (`sc_input.dat`) for the main program. When started the pre-processor will suggest default values for the parameters, but the user is able to type other input or choose from preselected items, see figure 3.2.

The first input is the name of the file containing the orientation of the grains represented by the three Euler angles, φ_1, Φ and φ_2 (Bunge 1982). The default is `random.dat` containing 5000 sets of Euler angles representing a random texture. The next input is the number of grains to be used in the calculations. The default is 200 grains, which is a reasonable number of grains for determination of τ_0 and the hardening coefficients. When these values have been determined a subsequent calculation with at least 5000 grains should be made to get the correct microstructural behaviour.

The next input is the overall strain increment ($\dot{\bar{\varepsilon}}_{11}$). The default value is 0.01% giving a maximum overshoot in τ of a few per cent. The first strain increment will be calculated automaticly from the initial Young's moduli insuring that there will

```
  ┌─────────────┐
 ( sc_ini.c )          The pre-processor.
  └─────────────┘

┌───────────────┐
│ Input data file│      The name of the file containing the grain orientations
│ <random.dat>  │      represented by their three Euler angles in degrees.
└───────────────┘

┌───────────────┐
│ How many      │
│ grains are to │      The number of grains to be used in the calculations.
│ be used       │
│    <200>      │
└───────────────┘
```

Overall strain increment $\dot{\bar{\varepsilon}}_{11}$ <0.01%>

Overall strain increment.

Control parameter $< \bar{\varepsilon}_{11} >$

Default is $\bar{\varepsilon}_{11}$. Other options are $\bar{\sigma}_{11}$ and $\bar{\varepsilon}_{11}^P$.

Maximum deformation <3.0%>

The maximum deformation: $\bar{\varepsilon}_{11} \Rightarrow < 3.0\% >$, $\bar{\sigma}_{11} \Rightarrow < 100\text{MPa} >$ and $\bar{\varepsilon}_{11}^P \Rightarrow < 3.0\% >$

Material <Aluminium>

Default is aluminium. Other options are copper, stainless steel, hybrid and ISO.

Hardening parameters

The default values for the hardening coefficients are dependent on the chosen material. The default values are shown in table 3.1.

Number of reflections <10>

The number of Bragg reflections. The reflections are listed in table 3.2

Angle interval $< 5° >$

The angular interval for selection of the grains that are within the reflections.

Number of half cycles <0>

The number of half cycles, where half cycles are defined as a loading to the maximum load and unloading to zero stress in the longitudinal direction, see figure 3.3.
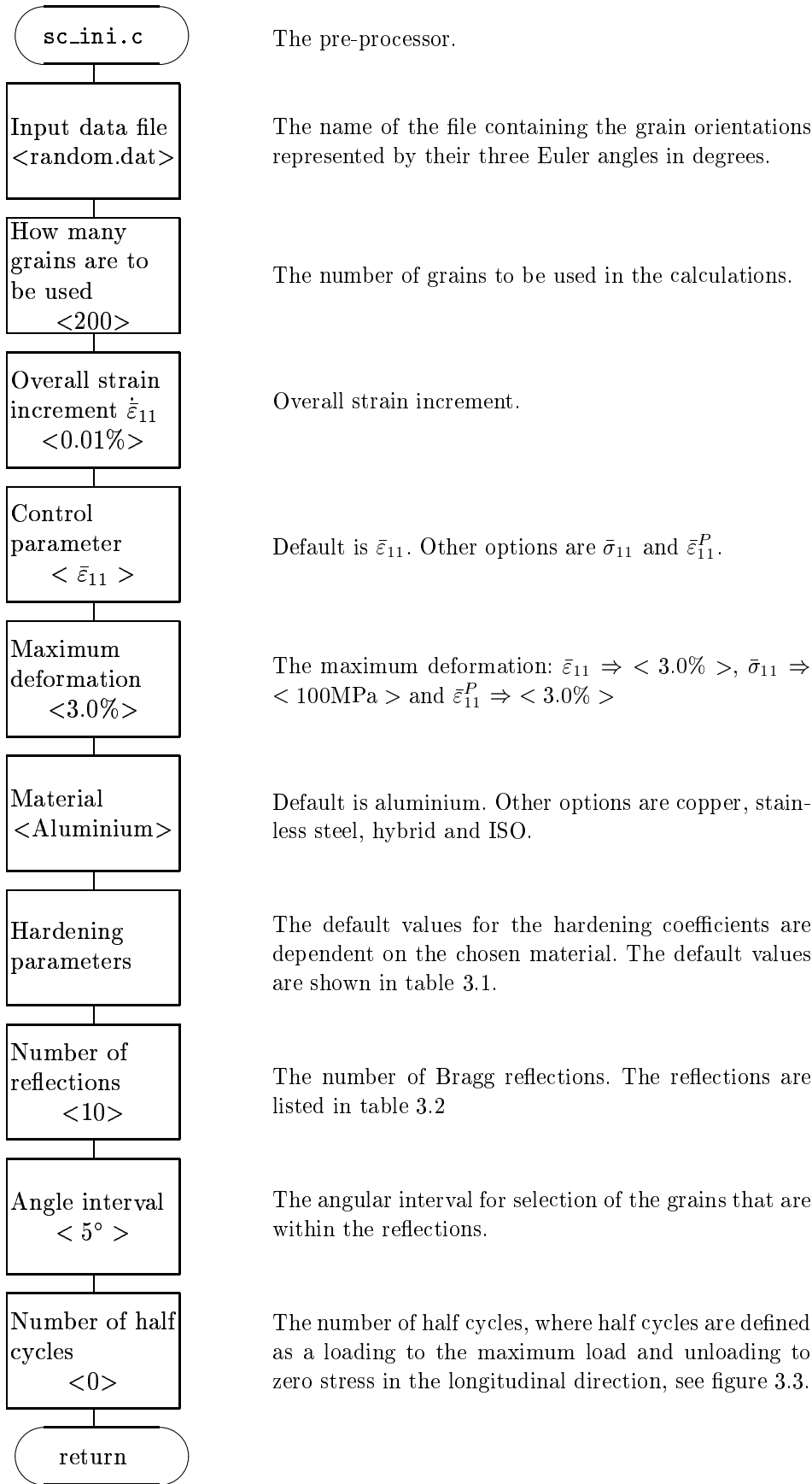
return

Figure 3.2. Flow chart for the pre-processor.

only be about 10 steps in the linear elastic regime. The following steps will be of 1/10 of the selected strain increment as the change in stiffness moduli is significant when the grains start to get active slip systems. The step size will gradually be increased to the selected value when all the grains have become plastic.

Then a selection of the controlling parameter for the maximum deformation has to be made. The default parameter is strain ($\bar{\varepsilon}_{11}$) and the default maximum strain is 3.0%. The other parameters that can be used are stress ($\bar{\sigma}_{11}$) or plastic strain ($\bar{\varepsilon}_{11}^P$).

The program has five pre-defined sets of material parameters; a fictive elastic isotropic material (ISO), aluminium, copper, a fictive material with the elastic anisotropy of copper and the Young's modulus of aluminium (hybrid) and stainless steel. The default values of the following parameter inputs will be related to the choice of material. The initial value of τ_0 and the hardening coefficients are the next inputs. τ_0, the final hardening coefficient and q must be larger than zero and q must not be equal to 1 as described in section 2. The default values of τ_0, hardening parameters and single crystal stiffnesses for the five materials are shown in table 3.1. The single crystal stiffnesses (C_{11}, C_{12} and C_{44}) must be provided as well as initial polycrystal stiffness (L_{ijkl}) and Young's modulus.

	τ_0	q	h_{final}	h_{ratio}	h_{exp}	C_{11}	C_{12}	C_{44}	$\frac{2\,C_{44}}{C_{11}-C_{12}}$
	MPa	—	MPa	—	—	GPa	GPa	GPa	—
ISO	65.0	1.01	140.0	20.0	200.0	501.0	198.0	151.5	1.00
Aluminium	10.9	1.01	40.0	5.0	61.0	108.2	61.3	28.5	1.22
Copper	15.0	1.01	250.0	20.0	400.0	168.4	121.4	75.4	3.21
Hybrid	10.9	1.01	40.0	5.0	61.0	92.2	66.5	41.2	3.21
St. steel	87.0	1.01	500.0	5.0	450.0	204.6	137.7	126.2	3.77

Table 3.1. Default values of initial critical resolved shear stress, hardening coefficients and single crystal stiffnesses (Dieter 1988, Ledbetter 1984)

	1	2	3	4	5	6	7	8	9	10
hkl	111	200	220	311	331	420	422	531	620	533
Multiplicity	8	6	12	24	24	24	24	48	24	24

Table 3.2. The reflections and their multiplicity

The next input is the number of reflections to determine stress and strain components for. The maximum number, 10, is the default and the sequence of reflections are shown in table 3.2. Lattice planes corresponding to forbidden reflections in the *fcc* lattice, higher-order lattice reflections and reflections composed of two families of lattice planes are ignored. Any number less than 10 will select the required number of reflections in the order shown in table 3.2.

The next input is the angle interval that selects the grains within the reflections. The default value is 5° which with 5000 grains will ensure a reasonable number of grains within the reflections. The number of grains within $\pm\alpha°$ of a direction in a material with random texture are determined by

$$N_{hkl}^{\alpha} = \frac{1}{2}\left(1 - \cos\alpha\right) \times N \times M_{hkl} \tag{3.1}$$

where N is the total number of grains and M_{hkl} is the multiplicity of the *hkl* reflection.

The last input is the number of half load cycles where a half cycle means load to the specified maximum and then unload to zero, see figure 3.3. The default is
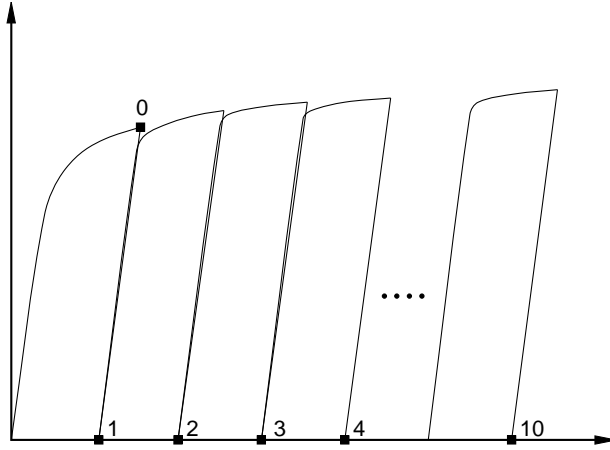
Figure 3.3. Definition of half cycles.

0, meaning loading to the selected maximum load without unloading. Choosing 1 half cycle the load will be taken to zero after the specified maximum is reached. The maximum number of half load cycles is 10. For N unloads the ith default unload value will be i/N times the maximum load ($\bar{\varepsilon}_{11}$, $\bar{\sigma}_{11}$ or $\bar{\varepsilon}_{11}^P$).

3.2   The main program (`sc_model.c`)

The following section is a detailed description of the main program. The general out-line of the subroutines in the main program is given in the flow charts in the figures, while the detailed description is given in the accompanying text. A list of all the subroutines are given in table A.1 in appendix A.

`main`

The `main` routine consists of the five subroutines: `define_arrays`, `array_ini`, `file_ini`, `cycle_loop` and `program_end`, see figure 3.4. Almost all variables are defined in the header making them global variables.

`define_array and array_ini`

The subroutine `define_array` defines the one, two and three dimensional pointer arrays used in the program and the subroutine `array_ini` initializes the material property arrays, such as stiffness moduli and constituent and matrix stress and strain arrays, see the figures 3.5 and 3.6. In this subroutine 10 ε_{11}^P levels are defined. At these levels the data needed for making pole plots of the number of active slip systems and the m-factor is printed in the `p_XX.dat` files, see page 17.

The main program.

Defines the size of the one, two and three dimensional pointer arrays, see figure 3.5.

Reads the input file from the pre-processor and defines the pointer arrays that are dependent on the user input. Initializes the arrays, see figure 3.6.

Opens all the output files, see figure 3.7.

Loop over the deformation cycles, see figure 3.8.

Frees all the allocated memory, see figure 3.9.

Figure 3.4. Flow chart for the main program.

Definition of one, two and three dimensional pointer arrays.

Reads the input file sc_ini.dat.

Definition of the pointer arrays that depend on the user input.
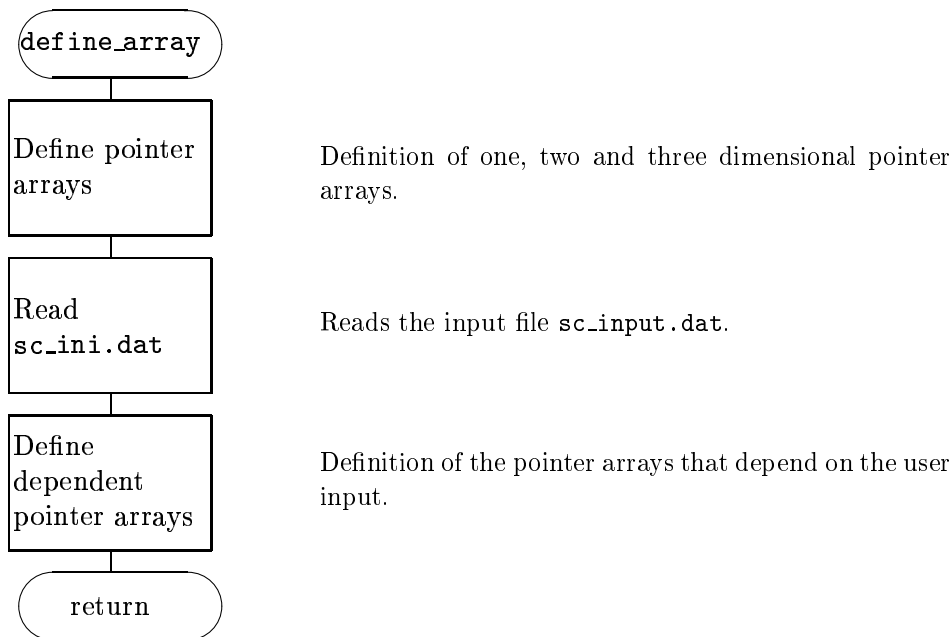
Figure 3.5. Flow chart for the subroutine define_array.

array_ini	
Define $\boldsymbol{n}$ and $\boldsymbol{m}$	Definition of the slip plane normals ($\boldsymbol{n}$) and slip directions ($\boldsymbol{m}$).
Define material properties	Definition of the load levels for pole plots.
Define inverse pole plot prints	Definition of the material parameters. Other materials can be added by copying the section with the material definitions.
Initialize ε, ε^P, σ, σ_c and $\dot{\sigma}_c$	Initialize the arrays.
return	

Figure 3.6. Flow chart for the subroutine `array_ini`.

file_ini

The subroutine `file_ini`, see figure 3.7, initializes the output files listed in the following. Examples of the output files are shown in appendix C.

- `self.dat`
  Containing general information of the calculations: $\bar{\varepsilon}_{11}$, $\bar{\varepsilon}_{22}$, $\bar{\varepsilon}_{33}$, $\bar{\varepsilon}_{23}$, $\bar{\varepsilon}_{31}$, $\bar{\varepsilon}_{12}$, $\bar{\varepsilon}_{11}^P$, $\bar{\sigma}_{11}$, $\bar{\sigma}_{22}$, $\bar{\sigma}_{33}$, $\bar{\sigma}_{23}$, $\bar{\sigma}_{31}$, $\bar{\sigma}_{12}$, $\b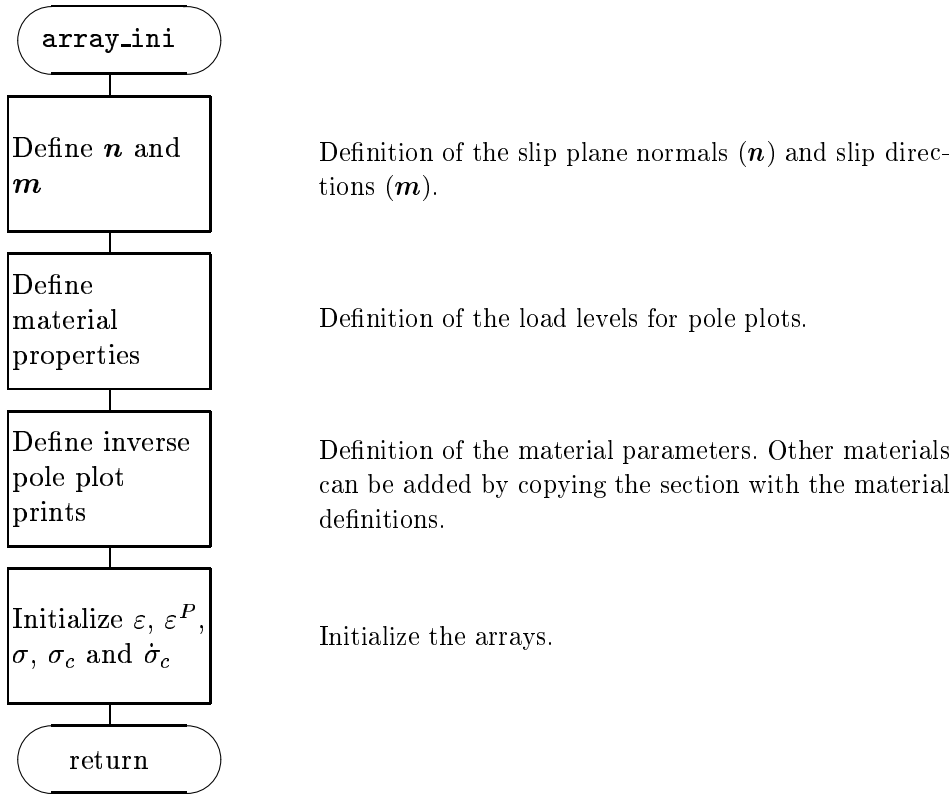ar{\nu}$, ν_{12}, ν_{13}, # of plastic grains, max # of active systems, max overshoot, # of grains over 1% overshoot, # of grains over 3% overshoot, average Taylor factor and # of grains with 1, 2, 3, 4, 5, 6, 7 and 8 active slip systems.

- `str_dir.dat`
  Containing stresses, elastic strains and the number of grains for the selected reflections parallel to the tensile axis (direction 1): $\bar{\varepsilon}_{11}$, ε_{EQ}^P, $\bar{\sigma}_{11}$, # of grains in reflection # 1, $\varepsilon_{11}^{E,ref\#1}$, $\sigma_{11}^{ref\#1}$, # of grains in reflection # 2, $\varepsilon_{11}^{E,ref\#2}$, $\sigma_{11}^{ref\#2}$, . . .

- `str_nor.dat` and `str_tra.dat`
  The same as `str_dir.dat` but for the normal (2) and transverse (3) directions.

- `stddev1.dat`
  Containing the standard deviation of the stresses and elastic strains for the selected reflections parallel to the tensile axis (direction 1): $\bar{\varepsilon}_{11}$, $\bar{\varepsilon}_{11}^P$, $\bar{\sigma}_{11}$, standard deviation of $\varepsilon_{11}^{E,ref\#1}$, standard deviation of $\sigma_{11}^{ref\#1}$, standard deviation of $\varepsilon_{11}^{E,ref\#2}$, standard deviation of $\sigma_{11}^{ref\#2}$, . . .

- `stddev2.dat` and `stddev3.dat`
  The same as `stddev1.dat` but for the normal (2) and transverse (3) directions.

- `p_XX.dat`
  Where XX are numbers from 01 to 10 corresponding to ten selected $\bar{\varepsilon}_{11}^P$ levels. These levels are defined in the sub-routine `array_ini`, see page 14. The files are containing Euler angles, Taylor factors and number of active slip systems for all the grains: φ_1, Φ, φ_2, # of active slip systems and the Taylor factor.

```
   ╭──────────╮
   │ file_ini │
   ╰──────────╯
        │
  ┌──────────────┐
  │ Open output  │      Opens the output files.
  │ files        │
  └──────────────┘
        │
  ┌──────────────┐
  │ Print headers│      Prints the headers to the output files.
  │ in output    │
  │ files        │
  └──────────────┘
        │
   ╭──────────╮
   │  return  │
   ╰──────────╯
```
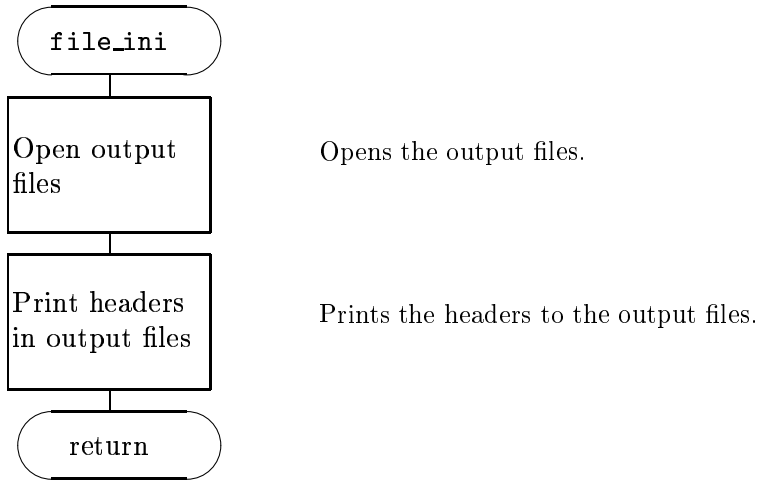
Figure 3.7. Flow chart for the subroutine `file_ini`.

cycle_loop

The subroutine `cycle_loop` is controlling the outer loop in the load sequence, see figure 3.8. The initial strain increment (step) is calculated from Young's modulus and τ_0. The following steps will be 1/10 of the selected step size. Once all the grains have become plastic the step size will be increased to 1/5, 1/2 and finally to the selected step size. This procedure ensures that the overshoot in τ does not increase above a few per cent.

Upon unloads the step size is $-1/10$ of the selected step size in the first 10 steps and then the load is reduced to just above zero in one step. The last 10 steps towards zero load is $-1/10$ of the initial step size. The reloading is made in the same way with 10 steps of 1/10 step size, then one large step to 1 step size below the unloading point and then 20 times 1/10 step size to ensure a smooth transition into the plastic region. This approach to zero load is only applicable when no slip systems are active during unloading. If any slip systems become active during unloading it is necesary to unload in increments of the normal step size.

program_end

The subroutine `program_end` clears the allocated memory and closes all files at the end of the program, see figure 3.9.

main_calc

For every step in the subroutine `cycle_loop` the subroutine `main_calc` is called. This subroutine calculates the new stresses and strains for the given strain increment, see figure 3.10. In the first step of the first iteration the file containing the Euler angles is opened. For every step a check file (`check.dat`) is opened and the progress of the calculations can be followed by monitoring this file. It contains

cycle_loop

Calculate first strain increment — Calculates the first strain increments from the Young's modulus and τ_0.

main_calc

Load ≥ limit

Iterates until the maximum load for this cycle is reached using the subroutine main_calc, see figure 3.10.

Unload

main_calc

$\bar{\sigma}_{11} \leq 0$

If there are specified unloads, the stress in the longitudinal direction is decreased to zero.

ncyc = max — Iterates until the maximum number of load cycles are reached.

return

Figure 3.8. Flow chart for the subroutine cycle_loop.

prog_end

Free allocated memory — Frees all the memory used by the program

return

Figure 3.9. Flow chart for the subroutine program_end.

the following parameters: step, iteration, $\bar{\varepsilon}_{11}$, number of plastic grains, maximum overshoot, grains with overshoot larger than 1%, grains with overshoot larger than 3%, maximum number of active slip systems in a grain, maximum deviation from last iteration and average Taylor factor.

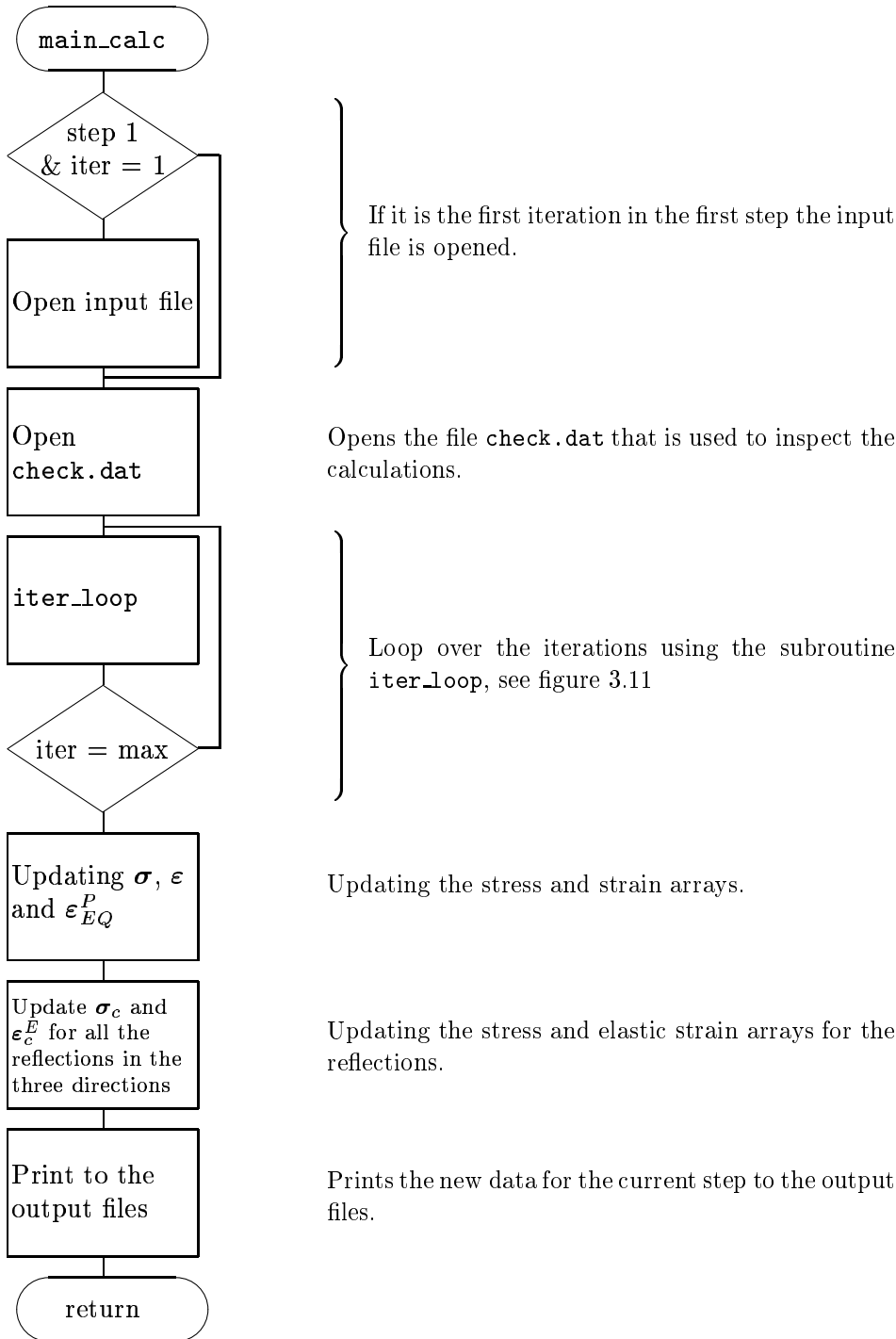|	
main_calc	
step 1 & iter = 1	If it is the first iteration in the first step the input file is opened.
Open input file	
Open check.dat	Opens the file `check.dat` that is used to inspect the calculations.
iter_loop	Loop over the iterations using the subroutine `iter_loop`, see figure 3.11
iter = max	
Updating $\boldsymbol{\sigma}$, ε and ε^P_{EQ}	Updating the stress and strain arrays.
Update $\boldsymbol{\sigma}_c$ and $\boldsymbol{\varepsilon}^E_c$ for all the reflections in the three directions	Updating the stress and elastic strain arrays for the reflections.
Print to the output files	Prints the new data for the current step to the output files.
return	

Figure 3.10. Flow chart for the subroutine `main_calc`.

`iter_loop`

In the subroutine `iter_loop`, see figure 3.11, the difference in all non-zero stiffness

If the difference from last iteration is less than 10^{-4} the iteration counter `iter` is set to the max value, and this iteration is the last one in this step.

Calculate the strain increment using the Young's modulus, the stiffness tensor and that $\bar{\sigma}_{11}$ is the only non-zero stress component.

If it is the last iteration in the step it is checked whether the data for a pole plot should be printed to the pole file at this load level.

And the arrays are reset to zero.

The constraint tensor $\boldsymbol{L}^*$ is calculated using the subroutine `lstar`, see figure 3.13.

Loop over all the grains using the subroutine `grain_loop`, see figure 3.15.

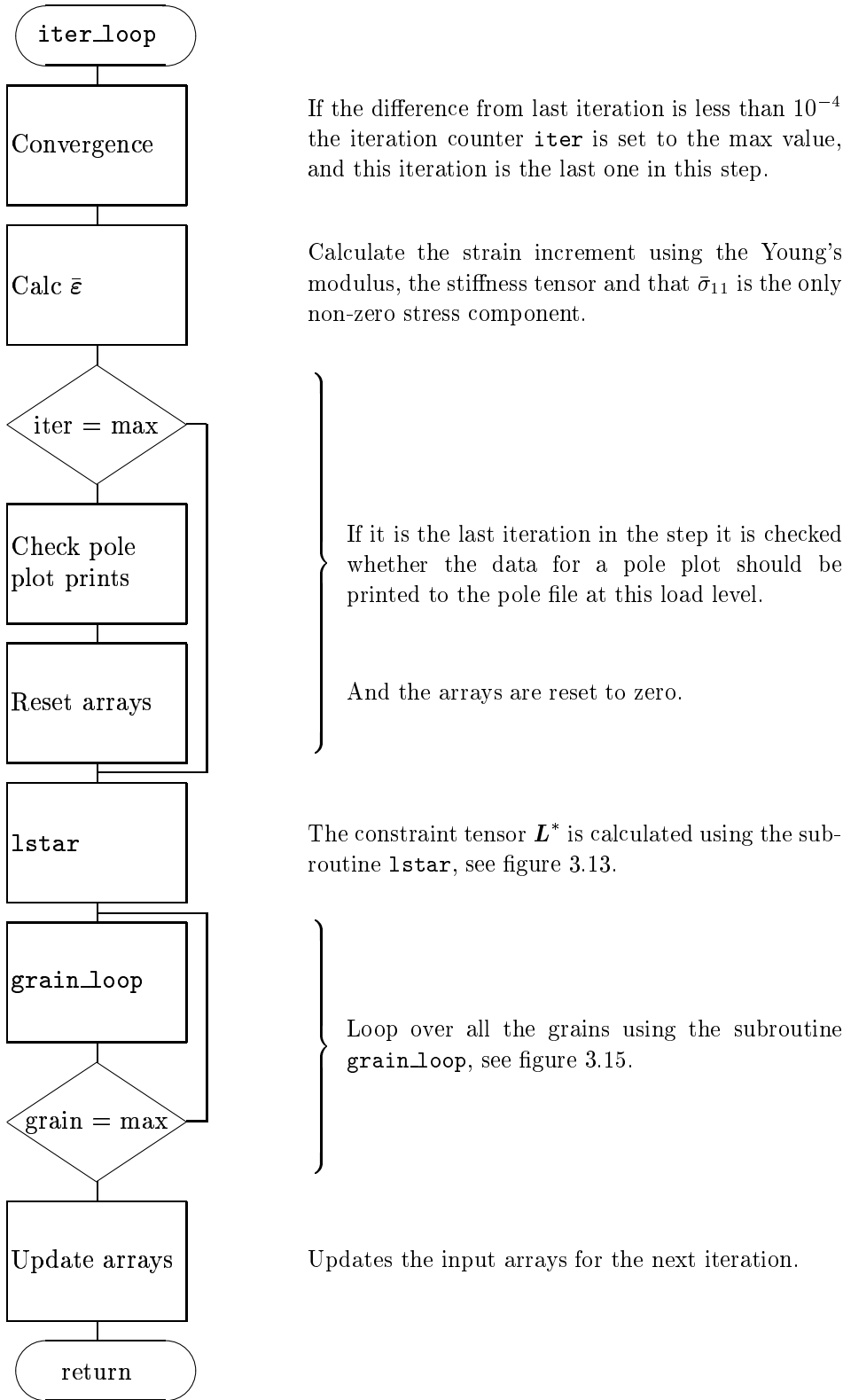Updates the input arrays for the next iteration.

Figure 3.11. Flow chart for the subroutine `iter_loop`.

tensor components from the last iteration is checked. For practical reasons non-zero components are defined as components that are larger than 5% of L_{1111}. If the difference is less than 10^{-4} the iteration counter will be set to the maximum value, and the loop will stop after this iteration. The maximum difference in stiffness tensor components is listed in output file `self.dat`.

For each iteration the total strain increment ($\bar{\varepsilon}_{ij}$) is calculated from the selected strain increment in the deformation direction ($\bar{\varepsilon}_{11}$) using the instantaneous overall stiffness tensor and the deformation type (uniaxial tension with $\bar{\sigma}_{11}$ as the only non-zero stress component).

If it is the last iteration in the step it is checked whether the data for a pole plot should be printed to the pole file at this load level and the arrays are reset to zero.

The constraint stiffness tensor, $\boldsymbol{L}^*$, is calculated using the subroutine `lstar` and the loop over the grains are performed. Then the arrays are updated.

step_func

The subroutine `step_func` determines the error function for the strain increment, see figure 3.12. The stress increment in the longitudinal direction $\dot{\bar{\sigma}}_{11}$ is calculated from the Young's modulus and the prescribed strain increment. The overall stiffness tensor and the stress increment are used to calculate the corresponding strain increment. The subroutine returns the difference between the calculated strain increment and the prescribed strain increment.

Copyies $\boldsymbol{L}_{old}$ and performs lower and upper decomposition using the auxiliary routine `ludcmp`, see page 37.

Performs backsubstitution with the input vector using the auxiliary routine `lubksb`, see page 37.

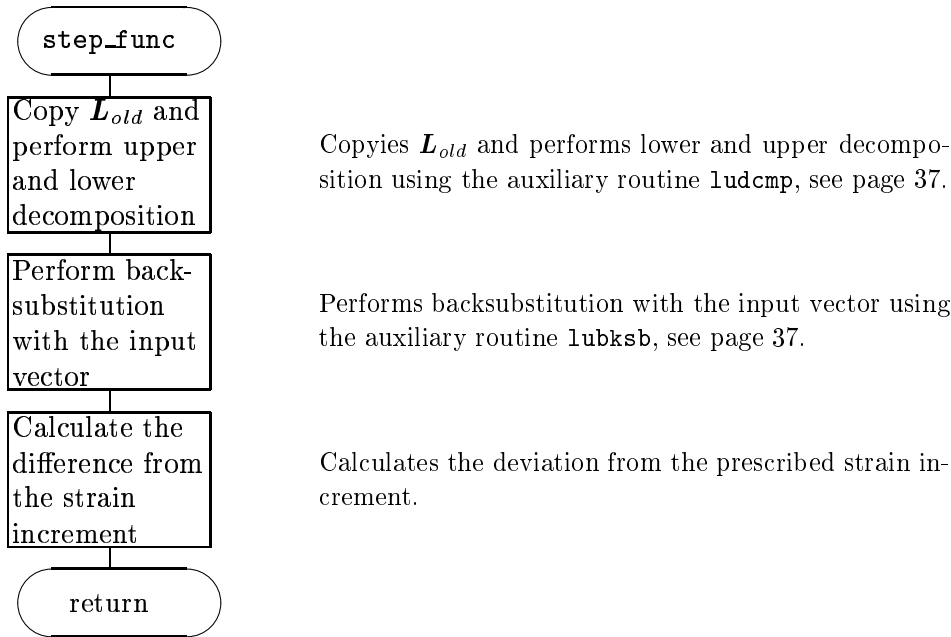Calculates the deviation from the prescribed strain increment.

Figure 3.12. Flow chart for the subroutine `step_func`.

lstar

The subroutine `lstar` calculates Hill's constraint tensor, see figure 3.13. The notation of the overall stiffness tensor is shifted to the normal fourth order tensor notation. Using the subroutines `quad2d` and `kneer` the double integral defining the Λ (see equation 2.13) tensor in the expression of the constraint tensor is calculated. The notation is shifted back to the orthonormal notation and the constraint tensor is calculated.
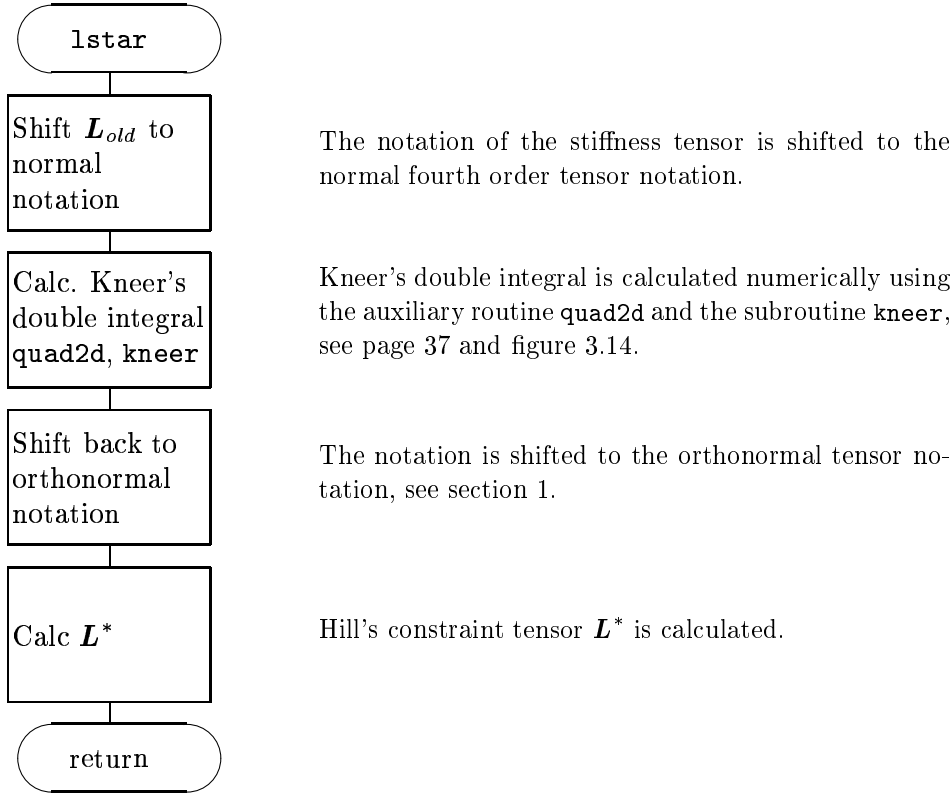
```
        ┌─────────────┐
        │    lstar    │
        └─────────────┘
      ┌───────────────┐
      │ Shift L_old to│        The notation of the stiffness tensor is shifted to the
      │ normal        │        normal fourth order tensor notation.
      │ notation      │
      └───────────────┘
      ┌───────────────┐
      │ Calc. Kneer's │        Kneer's double integral is calculated numerically using
      │ double integral│       the auxiliary routine quad2d and the subroutine kneer,
      │ quad2d, kneer │        see page 37 and figure 3.14.
      └───────────────┘
      ┌───────────────┐
      │ Shift back to │        The notation is shifted to the orthonormal tensor no-
      │ orthonormal   │        tation, see section 1.
      │ notation      │
      └───────────────┘
      ┌───────────────┐
      │ Calc L*       │        Hill's constraint tensor L* is calculated.
      └───────────────┘
        ┌─────────────┐
        │   return    │
        └─────────────┘
```

Figure 3.13. Flow chart for the subroutine `lstar`.

kneer

The subroutine `kneer` calculates the value of Kneer's volume function for determination of the constraint tensor, see figure 3.14. The trigonometric factors k_i are determined and the inverse of the matrix U is calculated. U is determined and the function value is calculated.

Calculation of the trigonometric factors k_i.

Calculation of the inverse of the U matrix.

Inversion of the U matrix using the auxiliary routines `ludcmp` and `lubksb`, see page 37.
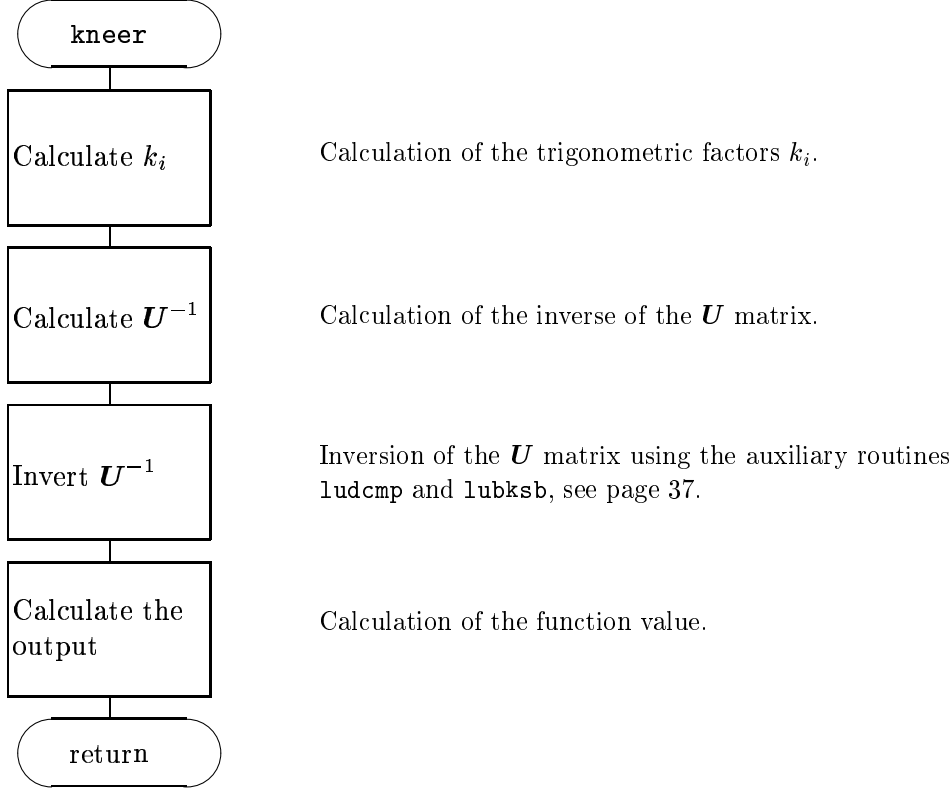
Calculation of the function value.

Figure 3.14. Flow chart for the subroutine `kneer`.

grain_loop

If it is the first iteration in the first cycle the subroutine `grain_loop`, see figure 3.15, reads the Euler angles from the input file and calculates the Cardanic angles. For all the other cycles and iterations the Cardanic angles are read directly from the memory. Then the rotated slip plane normals n, slip directions m and the Schmid factors μ, are calculated using the second order rotation matrix q. The rotated elastic stiffness tensor $\mathcal{L}_c$ is calculated using the fourth order rotation matrix R, and the hardening parameters h_γ and h_{ij} are determined.

The active slip systems are determined in the subroutine `det_act_sys` and in the subroutine `check_group` it is checked whether new slip systems become active or other gets inactive. The contribution from this grain is added to the overall stiffness tensor and the arrays are updated.

If it is the last iteration the lattice rotations are calculated in the subroutine `pole_print` and it is checked whether the data for a pole plot should be printed to the pole file at this load level.
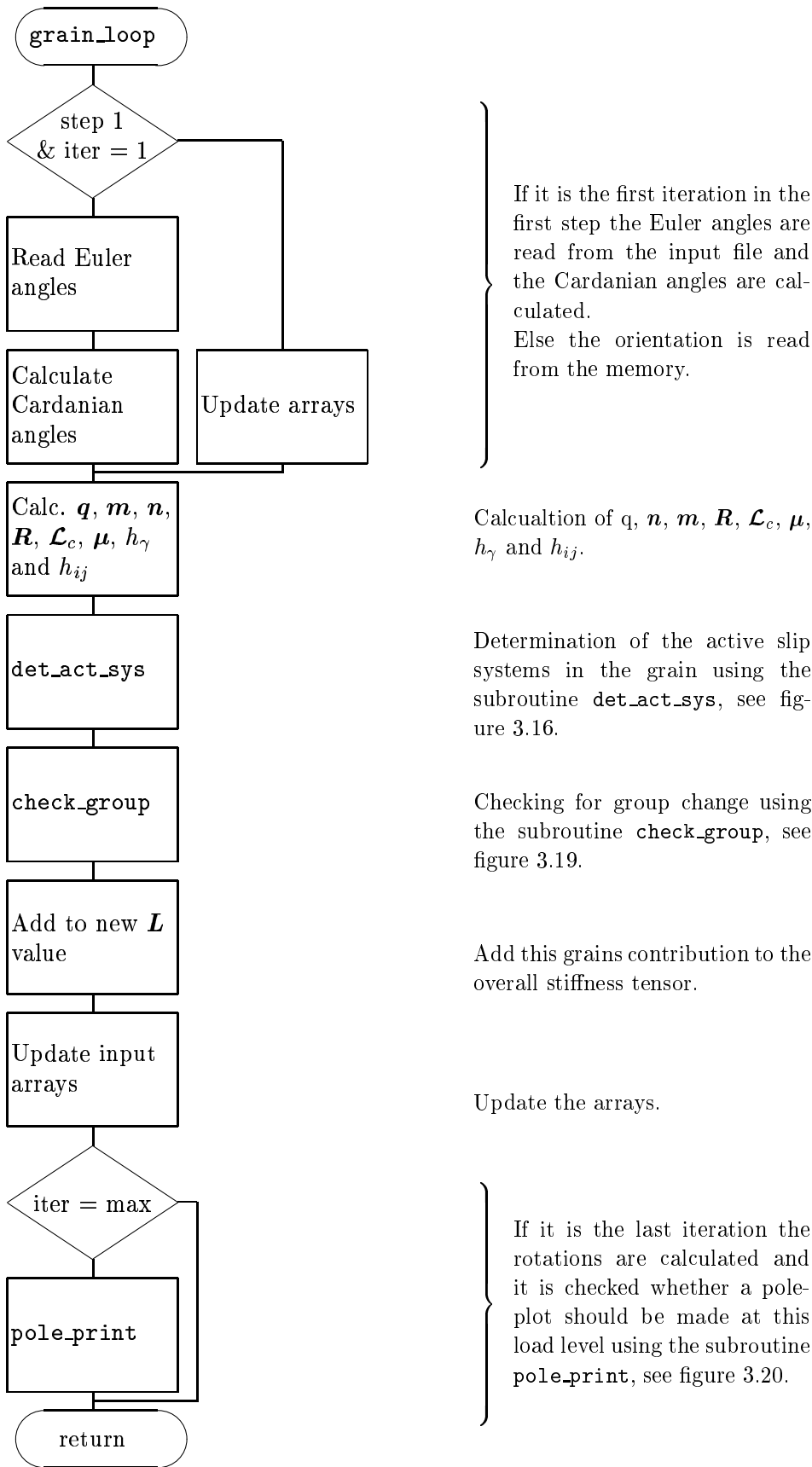
```
grain_loop
```

step 1 & iter = 1

Read Euler angles

Calculate Cardanian angles

Update arrays

If it is the first iteration in the first step the Euler angles are read from the input file and the Cardanian angles are calculated.
Else the orientation is read from the memory.

Calc. q, m, n, R, $\mathcal{L}_c$, μ, h_γ and h_{ij}

Calcualtion of q, n, m, R, $\mathcal{L}_c$, μ, h_γ and h_{ij}.

det_act_sys

Determination of the active slip systems in the grain using the subroutine det_act_sys, see figure 3.16.

check_group

Checking for group change using the subroutine check_group, see figure 3.19.

Add to new L value

Add this grains contribution to the overall stiffness tensor.

Update input arrays

Update the arrays.

iter = max

pole_print

If it is the last iteration the rotations are calculated and it is checked whether a pole-plot should be made at this load level using the subroutine pole_print, see figure 3.20.

```
return
```

Figure 3.15. Flow chart for the subroutine grain_loop.

det_act_sys

The subroutine det_act_sys determines the active slip systems in the grain for this iteration, see figure 3.16. The slip systems are placed in different groups whether they are active or inactive: Grains in group 1 and 2 are elastic and grains in group 3 have active slip systems. Group 2 is a intermediar stage where the resolved shear stress is equal to τ but the increment in the resolved shear stress is less than $\dot{\tau}$.

The group information from the last iteration is read from the memory and if there are active slip systems the subroutine calc_L_c is used to calculate the stiffness tensor for the grain. Hill's concentration tensor is calculated using the subroutine calc_A_c and the stress and strain increments for the grain are calculated. The slip rates are calculated and if any of them are negative this slip system is forced to be inactive and the calculations are repeated. If there are no active slip systems the stiffness tensor for the grain is equal to the elastic stiffness tensor.
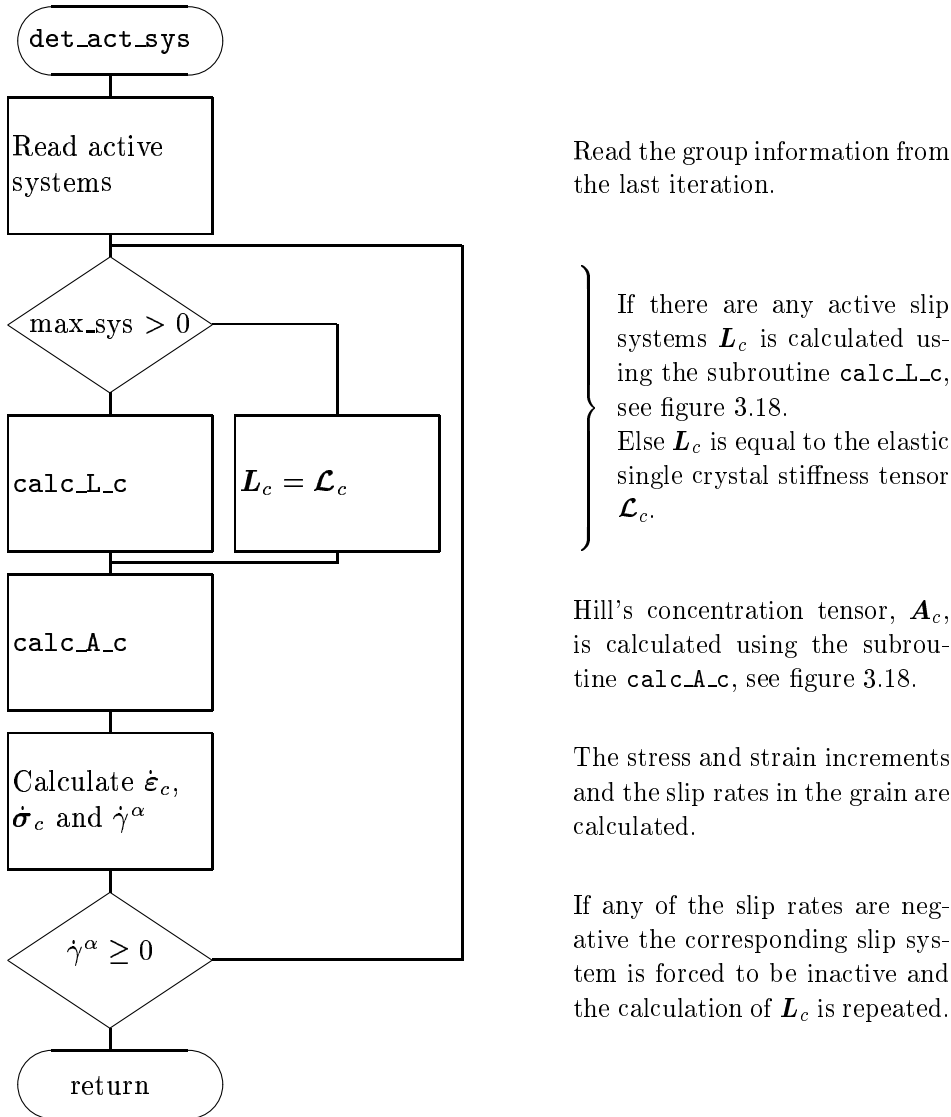
Read the group information from the last iteration.

If there are any active slip systems L_c is calculated using the subroutine calc_L_c, see figure 3.18.
Else L_c is equal to the elastic single crystal stiffness tensor $\mathcal{L}_c$.

Hill's concentration tensor, A_c, is calculated using the subroutine calc_A_c, see figure 3.18.

The stress and strain increments and the slip rates in the grain are calculated.

If any of the slip rates are negative the corresponding slip system is forced to be inactive and the calculation of L_c is repeated.

Figure 3.16. Flow chart for the subroutine det_act_sys.

`calc_L_c`

In the subroutine `calc_L_c`, see figure 3.17 the X matrix is calculated from the active slip systems. If the matrix is singular no further calculations are made. Else X is inverted and the $\boldsymbol{f}^i$ vectors that determine the relation between the slip rates and the strain increment are calculated. Then the grain stiffness tensor is calculated.
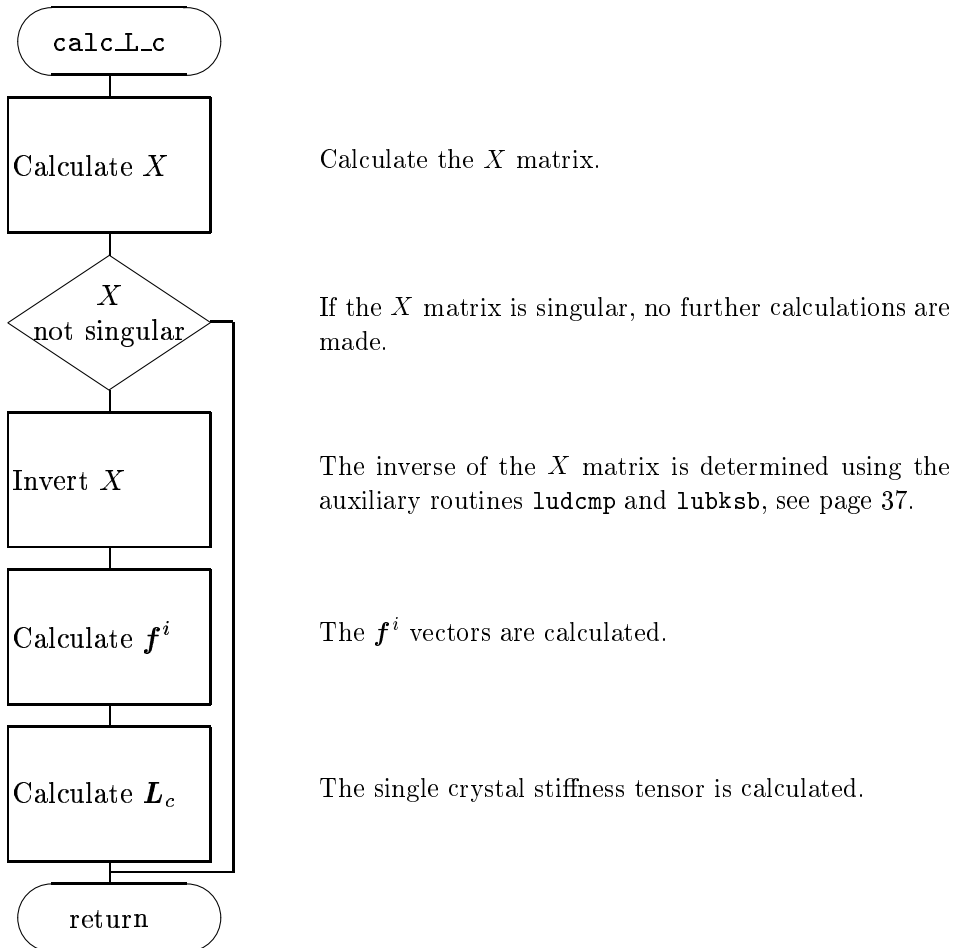
Calculate the X matrix.

If the X matrix is singular, no further calculations are made.

The inverse of the X matrix is determined using the auxiliary routines `ludcmp` and `lubksb`, see page 37.

The $\boldsymbol{f}^i$ vectors are calculated.

The single crystal stiffness tensor is calculated.

Figure 3.17. Flow chart for the subroutine `calc_L_c`.

calc_A_c

In the subroutine `calc_A_c` Hill's concentration tensor is calculated, see figure 3.18. First the constraint tensor and the stiffness tensor for the grain are multiplied and the resulting tensor is inverted and multiplied with the product of the constraint tensor and the overall stiffness tensor.
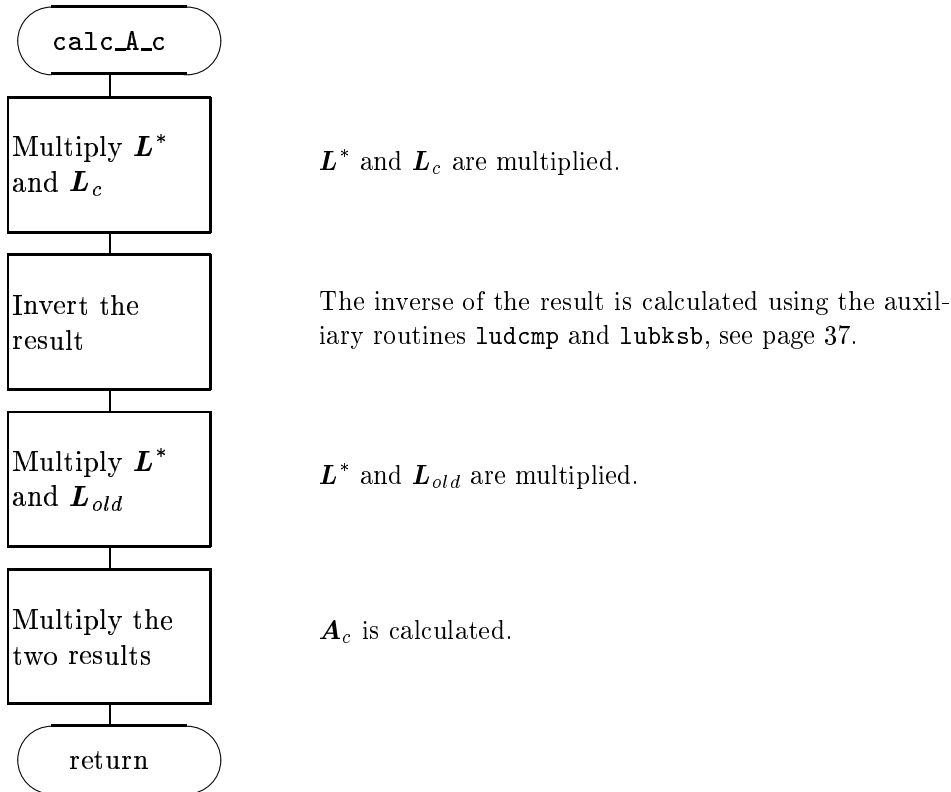

Flow chart	Description
calc_A_c	
Multiply $\boldsymbol{L}^*$ and $\boldsymbol{L}_c$	$\boldsymbol{L}^*$ and $\boldsymbol{L}_c$ are multiplied.
Invert the result	The inverse of the result is calculated using the auxiliary routines `ludcmp` and `lubksb`, see page 37.
Multiply $\boldsymbol{L}^*$ and $\boldsymbol{L}_{old}$	$\boldsymbol{L}^*$ and $\boldsymbol{L}_{old}$ are multiplied.
Multiply the two results	$\boldsymbol{A}_c$ is calculated.
return	

Figure 3.18. Flow chart for the subroutine `calc_A_c`.

check_group

In the subroutine `check_group` the new active slip systems are determined, see figure 3.19. The resolved shear stress, the increment in the resolved shear stress and the increment in τ are calculated for the 24 slip systems. The group change is checked as described in the flow chart for all the systems and the maximum overshoot is saved.
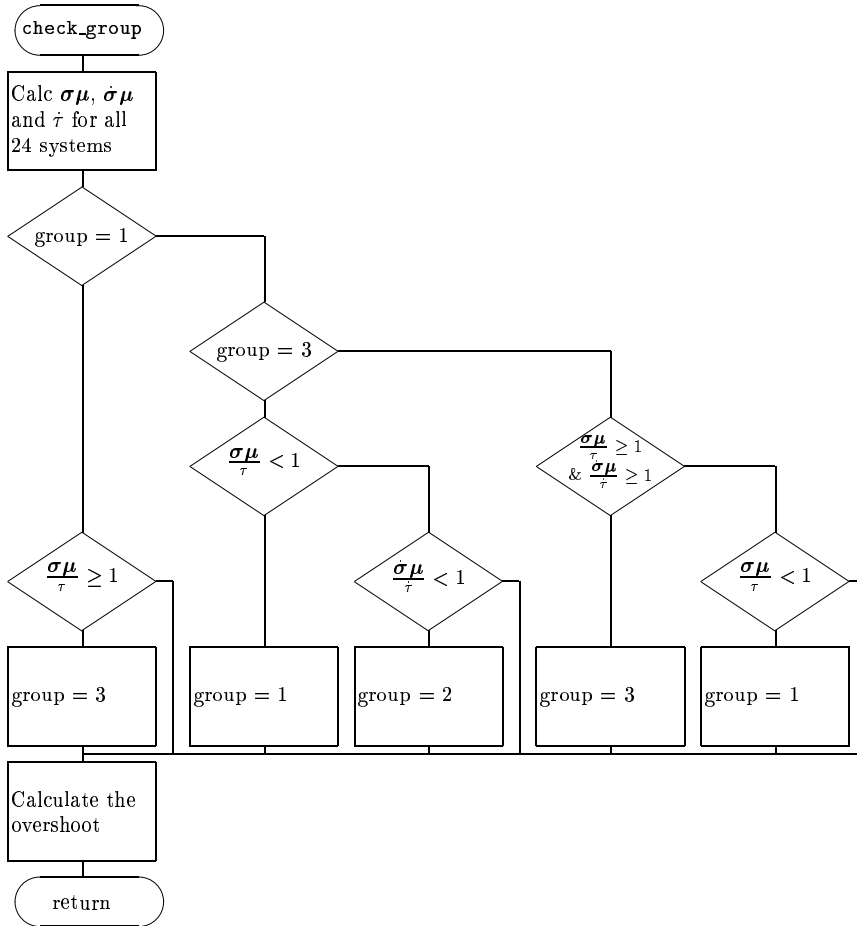
Figure 3.19. Flow chart for the subroutine `check_group`.

`pole_print`

In the subroutine `pole_print`, see figure 3.20, the lattice rotations in the grain and the new Euler angles are calculated and the arrays are updated. If a pole plot should be made at this load level the data is printed to the pole files. Then the elastic strains are calculated and using the subroutine `check_angle` it is checked whether the grain is within any of the reflections in one of the three directions.

Flow chart	Description
pole_print	
Calculate increment in Cardanic angles	The rotations are calculated.
Calculate new Euler angles	The new Euler angles are calculated.
Update arrays	The arrays are updated.
Print to pole files	If a pole plot should be made at this load level the data is printed to the pole files.
Calculate ε^E	The elastic strains are calculated.
`check_angle` for the three directions	The elastic strains and the stresses in the sub-sets of grains are written to the output files using the subroutine `check_angle`, see figure 3.21.
return	

Figure 3.20. Flow chart for the subroutine `pole_print`*.*

`check_angle`

The subroutine `check_angle` determines whether the grain is within one of the reflections, see figure 3.21. The direction vector is determined; longitudinal (1), normal (2) or transverse (3). For every reflection and for all the 48 possibilities in the *fcc* crystal structure the deviation between the orientation of the lattice plane normal of the reflection and the direction vector is determined. If the deviation angle is less than the limit the elastic lattice strain and the stress arrays are updated, and the subroutine is ended.

The direction is determined from the function input: dir = 1, nor = 2 and tra = 3.

The angle difference is calculated.

If Δv is smaller than the limit the arrays are updated, and no further calculations are made for this reflection.

Loop over the 48 directions.

Loop over the reflections.
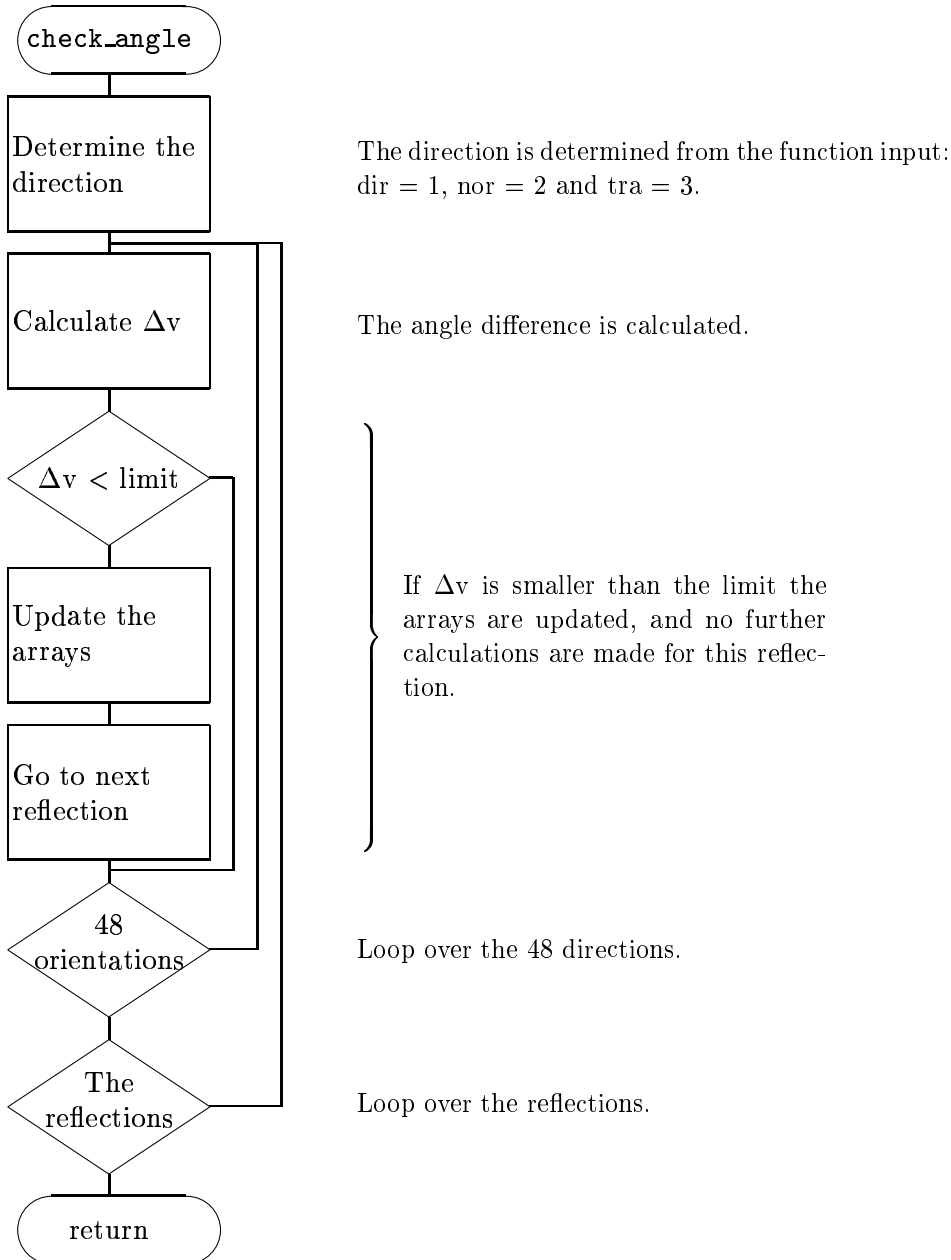
Figure 3.21. Flow chart for the subroutine `check_angle`.

4 Sample calculation

To enable the user to check the program after compilation the result of a sample calculation is presented in the following section. The parameters used in the pre-processor (sc_ini.c) are shown in table 4.1 and the output files are shown in appendix C. Selected output from the sample calculation is presented in the following figures.

Parameter	Value
Input file:	random.dat
Number of grains:	5000 grains
Strain increment:	0.01%
Control parameter:	$\bar{\varepsilon}_{11}^P$
Maximum deformation:	5.0%
Material:	Aluminium
Initial hardening coefficient:	$\tau_0 = 10.9$ MPa
Latent/self-hardening ratio:	$q = 1.01$
Final hardening coefficient:	$h_{final} = 40$MPa
Hardening ratio:	$h_{ratio} = 5$
Exponential hardening coefficient:	$h_{exp} = 61$
Number of reflections:	10
The angle interval:	$5°$
Number of unloads	1

Table 4.1. Input parameters for the sample calculation.

(a) The overall stress.   (b) The active slip system number.
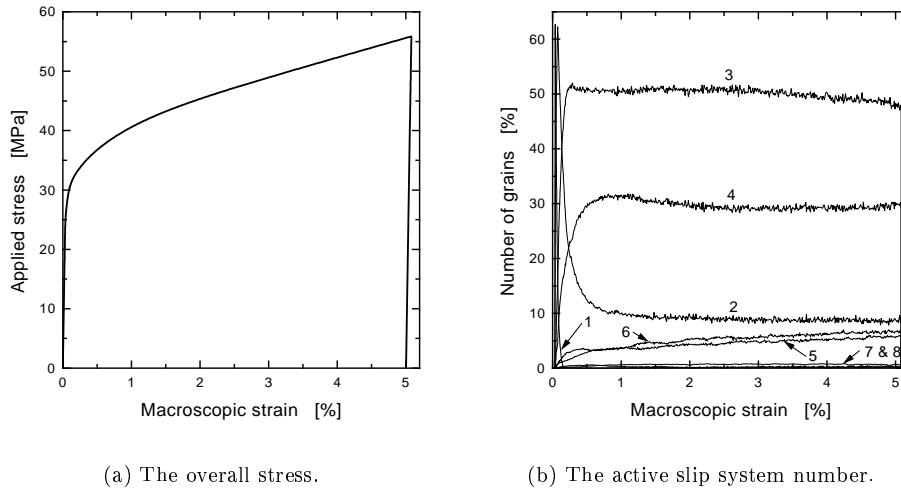
Figure 4.1. Overall stress and active slip system versus the macroscopic strain.

The parameters shown in figure 4.1 are the overall stress-strain response of the model, and the active slip system number in the grains versus the macroscopic strain. The overall stress-strain response in figure 4.1(a) shows that the exponential decreasing hardening law enables the model to describe a smooth transition from the elastic to the plastic region. The active slip system number in figure 4.1(b) shows that the dominating deformation mode is one, in which grains have three

active slip systems. After the transition zone approximately 50% of the grains accommodate the deformation by three active slip systems, while less than 10% of the grains utilize the general five active slip systems used in a Taylor model. The data for the unload to zero stress is only shown in figure 4.1(a) as it would make the following figures less clear.
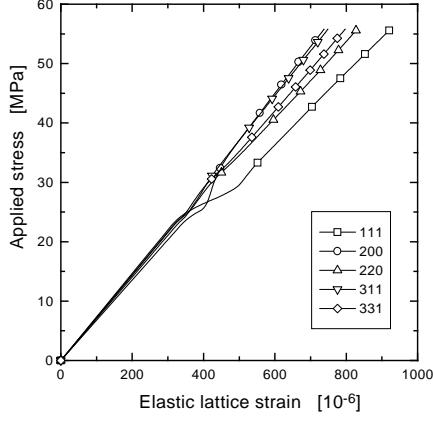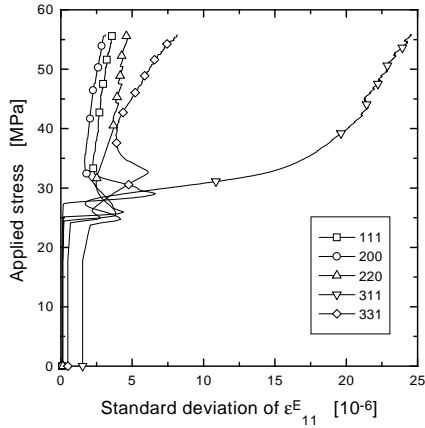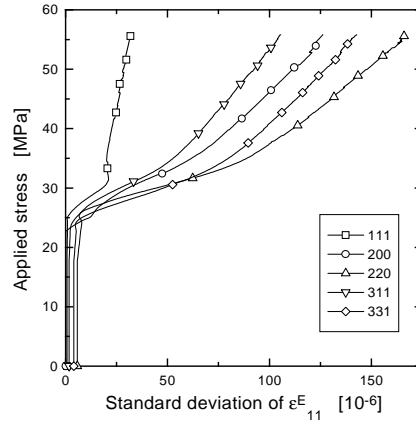

(a) Parallel to the tensile axis.
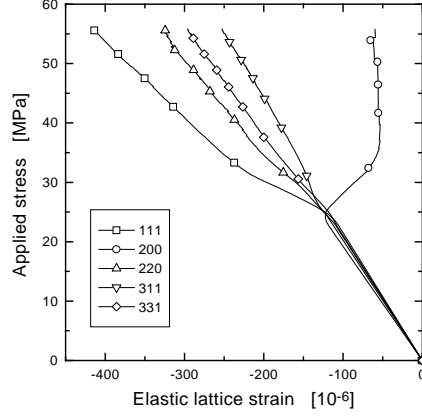
(b) Perpendicular to the tensile axis.

Figure 4.2. The applied load versus the elastic strain in the reflections.
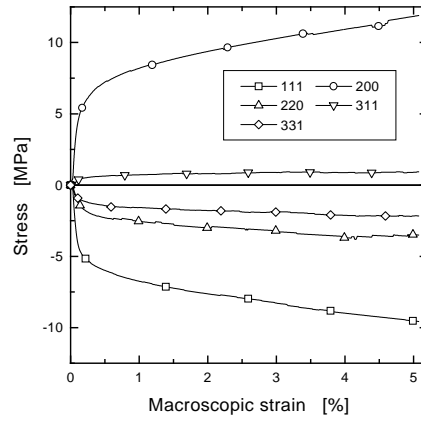

(a) Parallel to the tensile axis.

(b) Perpendicular to the tensile axis.

Figure 4.3. The applied load versus the standard deviation of the elastic strain in the reflections.

The figures 4.2 to 4.5 show the parameters with interest to neutron diffraction measurements; elastic lattice strain response and stress response for the grain subsets, and the standard deviation of these parameters. The elastic lattice strain results are presented with the independent variable (the applied load) along the y axis and the results along the x axis - in order to approach the presentation in a conventional stress-strain curve, see figure 4.2 and 4.3.
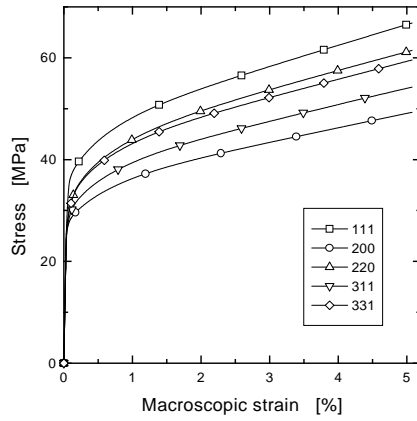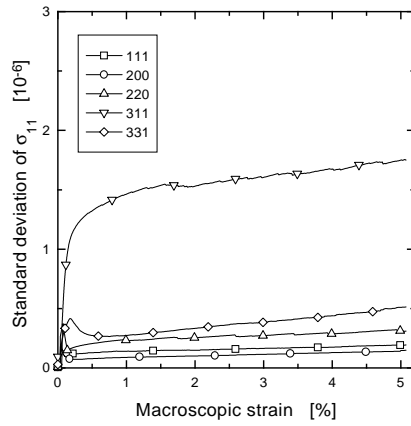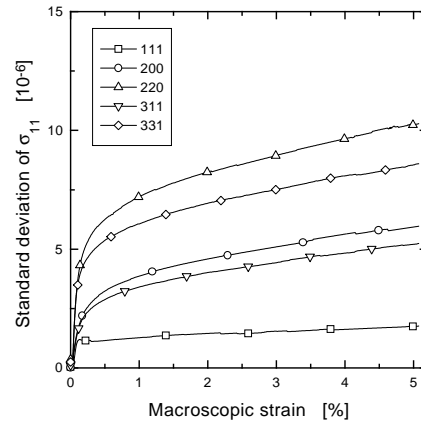
(a) Parallel to the tensile axis. (b) Perpendicular to the tensile axis.

Figure 4.4. The average stress in the reflections versus the macroscopic strain.

(a) Parallel to the tensile axis. (b) Perpendicular to the tensile axis.

Figure 4.5. The standard deviation of the stress in the reflections versus the macroscopic strain.

The results from the `p_XX.dat` files are used to present the orientation dependence of the number of active slip systems and the Taylor factor as shown in the figures 4.6 and 4.7.

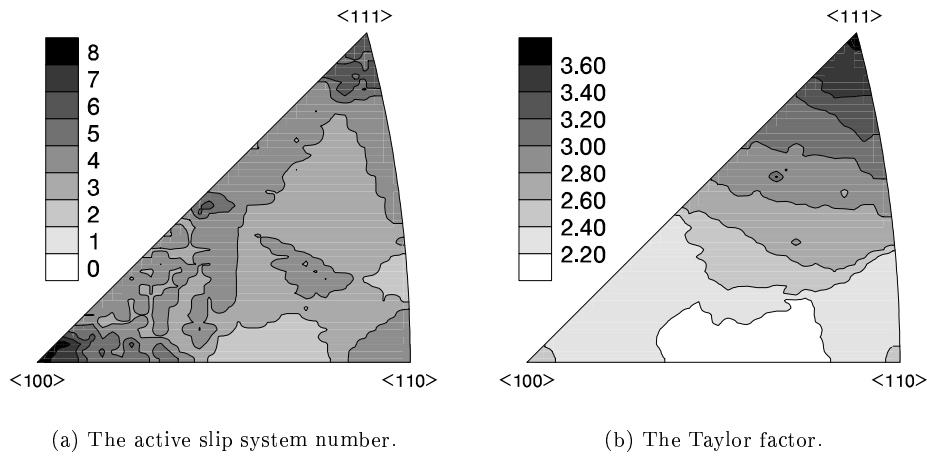(a) The active slip system number.

(b) The Taylor factor.

Figure 4.6. The orientation dependence of the active slip system number and the Taylor factor at 0.5% plastic deformation (`p_05.dat`*).*

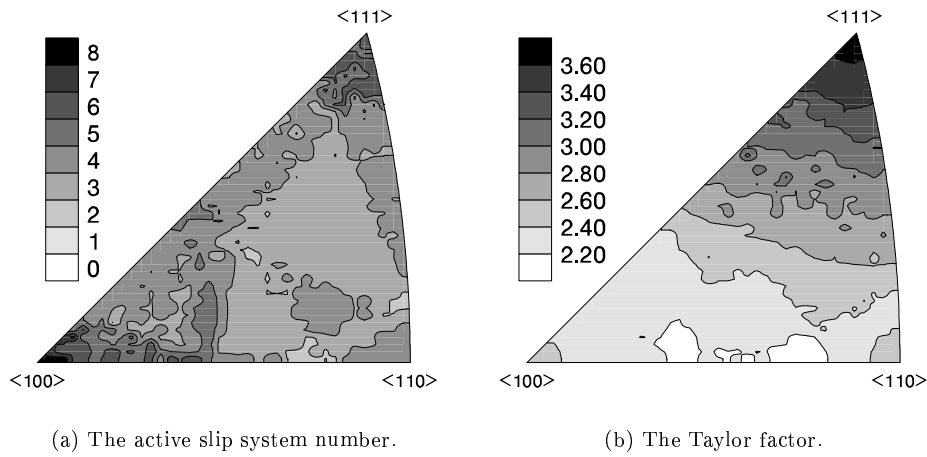(a) The active slip system number.

(b) The Taylor factor.

Figure 4.7. The orientation dependence of the number of active slip systems and the Taylor factor at 5% plastic deformation (`p_10.dat`*).*

5 Alteration and requirements

Due to the iterative nature of the self-consistent scheme the computing is rather time consuming. The calculation time for the example shown in section 4 was approximately 35000 CPU seconds on a Power Challenge R8000 Silicon Graphics work station and for shorter runs with 200 grains to about 3% deformation the calculation time is approximately 2000 CPU seconds.

The program is written in ANSI C and should be portable to any platform with an ANSI C compiler. The present implementation is compiled with the $-O3$ option for optimized compilation. The runtime memory requirements depend on the maximum grain number. For the calculation example the program uses approximately 4.8 Mb of memory and the output files are of approximately 3.9 Mb. Due to the memory requirements it is not possible to run the program on a DOS based computer without by-passing the 640 Kb limit.

It is recommended to run the program on a work station or a powerful PC due to the high calculation time. The calculation time may be reduced if the program is rewritten for multiple CPU's. As the calculation for each grain is made separately these calculations can be made at the same time. The effect will, of course, be largest when the number of grains is relatively high.

A   List of subroutines

Subroutine name	Description
def_array	Defines the one, two and three dimensional pointer arrays, see flow chart page 15.
array_ini	Reads the input file sc_input.dat and defines the arrays that are dependent on the user input. Initializes the arrays, see flow chart page 16.
file_ini	Opens all the output files, see flow chart 17.
cycle_loop	Loop over the load cycles, see flow chart page 18.
program_end	Frees allocated memory, see flow chart page 18.
main_calc	Calculates and updates the stresses and strains after each iteration, see flow chart page 19.
iter_loop	Loop over the iterations, see flow chart page 20.
step_func	Error function for determination of the strain increment, see flow chart page 21.
lstar	Calculates the constraint tensor, $\boldsymbol{L}^*$, see flow chart page 22.
kneer	Function used to determine Hill's constraint stiffness tensor, see flow chart page 23 (Hutchinson 1970).
grain_loop	Loop over the grains, see flow chart page 24.
det_act	Determines the active slip systems, the stress and strain increments and the slip rates in the grain, see flow chart page 25.
calc_L_c	Calculates instantaneous stiffness tensor for a grain, $\boldsymbol{L}_c$, see flow chart page 26.
calc_A_c	Calculates the Hill (Hill 1965a, Hill 1965b) concentration tensor, $\boldsymbol{A}_c$, see flow chart page 27.
check_group	Determines the active slip systems for the next iteration, see flow chart page 28.
pole_print	Prints the output to the pole files, see flow chart page 29.
check_angle	Determines whether the grain is within one of the selected reflections, see flow chart page 30.
dsqr	Calculates the square of a double.
isqr	Calculates the square of an integer.

From 'Numerical recipes in C'

nrerror	Standard error handler.
ivector	Allocates memory for a one dimensional integer array.
imatrix	Allocates memory for a two dimensional integer array.
vector	Allocates memory for a one dimensional double array.
matrix	Allocates memory for a two dimensional double array.
f3tensor	Allocates memory for a three dimensional double array.
free_ivector	Frees allocated memory for a one dimensional integer array.
free_imatrix	Frees allocated memory for a two dimensional integer array.
free_vector	Frees allocated memory for a one dimensional double array.
free_matrix	Frees allocated memory for a two dimensional double array.
free_f3tensor	Frees allocated memory for a three dimensional double array.
ludcmp	Lower and upper decomposition of a double matrix.
lubksb	Lower and upper back substitution for solving linear equations.
rtsec	Root finding with the secant method.
qgaus	Calculates the integral of a specified function.
quad2d	Calculates the double integral of a two dimensional function using qgaus.
f1, f2, yy1 and yy2	Auxiliary functions to quad2d determining the boundary conditions.

Table A.1. Subroutines used in the main program.

B   Nomenclature

The nomenclature used in the present work is mainly the one used in (Hutchinson 1970). Boldface lower case letters are used as symbols for second-order Cartesian tensors and fourth-order Cartesian tensors are represented by boldface upper case letters. The contracted product of two second-order tensors, $a_{ij}b_{ij}$, is written as $\boldsymbol{ab}$ and the inner product of two fourth-order tensors, $A_{ijmn}B_{mnkl}$, is written as $\boldsymbol{AB}$. The tensors are represented in an orthonormal dyadic base as described in (Pedersen 1995): The symmetric second-order tensors are represented by a vector and the original off-diagonal components are multiplied with $\sqrt{2}$ as shown in equation B.1.

$$\boldsymbol{a} = \begin{bmatrix} a_{11} & a_{22} & a_{33} & \sqrt{2}a_{23} & \sqrt{2}a_{31} & \sqrt{2}a_{12} \end{bmatrix} \tag{B.1}$$

The symmetric fourth-order tensors are represented by a matrix where the components in the upper right quadrant and in the lower left quadrant are multiplied with $\sqrt{2}$ and the components in the lower right quadrant is multiplied with 2 as shown in equation B.2.

$$\boldsymbol{A} = \begin{bmatrix} A_{1111} & A_{1122} & A_{1133} & \sqrt{2}A_{1123} & \sqrt{2}A_{1131} & \sqrt{2}A_{1112} \\ A_{2211} & A_{2222} & A_{2233} & \sqrt{2}A_{2223} & \sqrt{2}A_{2231} & \sqrt{2}A_{2212} \\ A_{3311} & A_{3322} & A_{3333} & \sqrt{2}A_{3323} & \sqrt{2}A_{3331} & \sqrt{2}A_{3312} \\ \sqrt{2}A_{2311} & \sqrt{2}A_{2322} & \sqrt{2}A_{2333} & 2A_{2323} & 2A_{2331} & 2A_{2312} \\ \sqrt{2}A_{3111} & \sqrt{2}A_{3122} & \sqrt{2}A_{3133} & 2A_{3123} & 2A_{3131} & 2A_{3112} \\ \sqrt{2}A_{1211} & \sqrt{2}A_{1222} & \sqrt{2}A_{1233} & 2A_{1223} & 2A_{1231} & 2A_{1212} \end{bmatrix} \tag{B.2}$$

This definition implies that the unity tensor satisfying $\boldsymbol{I} = \boldsymbol{A}^{-1}\boldsymbol{A}$, defined as $I_{ijkl} = 1/2\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right)$, renders a matrix with one's in all the diagonal elements.

C Examples of output files

This is the contents of the output file self.dat.

The number of grains is: 5000
The strain increment is: 0.010
The maximum strain is: 3.100
The angle interval is: 5.0
The chosen material is: Aluminium.
The chosen tau0 is: 10.900 MPa
The Hardening law is Hill hardening
The ratio between latent and primary hardening is: 1.010
The final hardening coefficient is: 40.
The ratio of initial to final hardening coefficient is: 5.0.
The exponential parameter in the hardening law is: 61.0.

e_1	e_2	e_3	e_4	e_5	e_6	ep_{P1}	s_1	s_2	s_3	s_4	s_5	s_6	ny	ny12	ny13	PlGr	Act	Over	O1	O3	Dev	Redu	Gam	h_m	M	a1	a2	a3	a4	a5	a6	a7	a8
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.000	0.000	0.000	0.000	0.000	0.000	0.3462	0.3462	0.3462	0	0	0.000	0	0	0.00e+00	0	0	200.0	0.000	0	0	0	0	0	0	0	0
0.0250	−0.0086	−0.0087	0.0000	0.0000	−0.0001	0.0000	17.744	0.000	0.000	0.000	0.000	0.001	0.3462	0.3456	0.3468	0	0	0.000	0	0	3.20e−05	0	0	200.0	0.000	0	0	0	0	0	0	0	0
0.0260	−0.0090	−0.0090	0.0000	0.0000	−0.0001	0.0000	18.453	0.000	0.000	0.000	0.000	0.001	0.3462	0.3456	0.3469	0	0	0.000	0	0	1.61e−06	0	0	200.0	0.000	0	0	0	0	0	0	0	0
0.0270	−0.0093	−0.0094	0.0000	0.0000	−0.0001	0.0000	19.163	0.000	0.000	0.000	0.000	0.001	0.3462	0.3456	0.3469	0	0	0.000	0	0	8.38e−08	0	0	200.0	0.000	0	0	0	0	0	0	0	0
0.0280	−0.0097	−0.0097	0.0000	0.0000	−0.0001	0.0000	19.873	0.000	0.000	0.000	0.000	0.001	0.3462	0.3456	0.3469	0	0	0.000	0	0	1.40e−08	0	0	200.0	0.000	0	0	0	0	0	0	0	0
0.0290	−0.0100	−0.0101	0.0000	0.0000	−0.0001	0.0000	20.582	0.000	0.000	0.000	0.000	0.001	0.3462	0.3456	0.3469	0	0	0.000	0	0	2.89e−09	0	0	200.0	0.000	0	0	0	0	0	0	0	0
0.0300	−0.0104	−0.0104	0.0000	0.0000	−0.0001	0.0000	21.292	0.000	0.000	0.000	0.000	0.001	0.3462	0.3456	0.3469	0	0	0.000	0	0	6.32e−10	0	0	200.0	0.000	0	0	0	0	0	0	0	0
0.0310	−0.0107	−0.0108	0.0000	0.0000	−0.0001	0.0000	22.002	0.000	0.000	0.000	0.000	0.001	0.3462	0.3456	0.3469	0	0	0.000	0	0	1.40e−10	0	0	200.0	0.000	0	0	0	0	0	0	0	0
0.0320	−0.0111	−0.0111	0.0000	0.0000	−0.0001	0.0000	22.712	0.000	0.000	0.000	0.000	0.001	0.3462	0.3456	0.3469	0	0	0.000	0	0	3.10e−11	0	0	200.0	0.000	0	0	0	0	0	0	0	0
0.0330	−0.0114	−0.0115	0.0000	0.0000	−0.0001	0.0002	23.316	0.000	0.000	0.000	0.000	0.001	0.3691	0.3666	0.3717	654	1	1.454	197	0	2.61e−05	0	0	200.0	2.017	654	0	0	0	0	0	0	0
0.0340	−0.0118	−0.0119	0.0000	0.0000	−0.0001	0.0005	23.772	0.000	0.000	0.000	0.000	0.001	0.4012	0.4014	0.4012	1691	2	2.826	873	0	5.84e−05	0	0	200.0	2.046	1549	142	0	0	0	0	0	0
0.0350	−0.0122	−0.0123	0.0000	0.0000	−0.0001	0.0010	24.145	0.000	0.000	0.000	0.000	0.001	0.4191	0.4168	0.4215	2370	2	2.826	1268	0	5.35e−05	0	0	199.9	2.070	2107	263	0	0	0	0	0	0
0.0360	−0.0127	−0.0127	0.0000	0.0000	−0.0001	0.0016	24.455	0.000	0.000	0.000	0.000	0.001	0.4330	0.4294	0.4367	2979	2	2.825	1595	0	6.06e−05	0	0	199.8	2.093	2598	381	0	0	0	0	0	0
0.0370	−0.0131	−0.0132	0.0001	0.0000	−0.0001	0.0022	24.710	0.000	0.000	0.000	0.000	0.001	0.4447	0.4412	0.4485	3550	2	2.824	1885	0	8.11e−05	0	0	199.7	2.117	3009	541	0	0	0	0	0	0
0.0380	−0.0136	−0.0136	0.0001	0.0000	−0.0001	0.0029	24.933	0.000	0.000	0.000	0.000	0.001	0.4517	0.4485	0.4551	3896	2	2.823	2047	0	8.15e−05	0	0	199.5	2.133	3135	761	0	0	0	0	0	0
0.0390	−0.0140	−0.0141	0.0001	0.0000	−0.0001	0.0037	25.135	0.000	0.000	0.000	0.000	0.001	0.4562	0.4541	0.4585	4114	2	2.822	2146	0	7.72e−05	0	0	199.4	2.145	3131	983	0	0	0	0	0	0
0.0400	−0.0145	−0.0146	0.0001	0.0000	−0.0001	0.0044	25.323	0.000	0.000	0.000	0.000	0.001	0.4593	0.4561	0.4626	4253	2	2.822	2204	0	7.57e−05	0	0	199.3	2.154	3049	1204	0	0	0	0	0	0
0.0410	−0.0149	−0.0150	0.0001	−0.0001	−0.0002	0.0052	25.498	0.000	0.000	0.000	0.000	0.001	0.4622	0.4598	0.4646	4383	4	2.821	2248	0	8.61e−05	0	0	199.1	2.163	2955	1416	11	1	0	0	0	0
0.0420	−0.0154	−0.0155	0.0001	−0.0001	−0.0002	0.0060	25.662	0.000	0.000	0.000	0.000	0.001	0.4643	0.4602	0.4686	4469	4	2.820	2268	0	8.65e−05	0	0	198.9	2.172	2829	1609	25	6	0	0	0	0
0.0430	−0.0159	−0.0160	0.0001	−0.0001	−0.0002	0.0067	25.819	0.000	0.000	0.000	0.000	0.001	0.4662	0.4625	0.4699	4537	4	2.819	2285	0	8.51e−05	0	0	198.8	2.179	2734	1723	57	23	0	0	0	0
0.0440	−0.0163	−0.0164	0.0001	−0.0001	−0.0002	0.0076	25.967	0.000	0.000	0.000	0.000	0.001	0.4678	0.4627	0.4730	4593	4	2.819	2304	0	9.01e−05	0	0	198.6	2.186	2632	1834	86	41	0	0	0	0
0.0450	−0.0168	−0.0169	0.0001	−0.0001	−0.0002	0.0084	26.109	0.000	0.000	0.000	0.000	0.001	0.4692	0.4635	0.4751	4648	4	2.818	2315	0	8.80e−05	0	1	198.4	2.194	2549	1927	116	56	0	0	0	0
0.0460	−0.0172	−0.0174	0.0001	−0.0001	−0.0002	0.0092	26.245	0.000	0.000	0.000	0.000	0.001	0.4704	0.4651	0.4759	4681	7	2.817	2322	0	9.20e−05	0	1	198.3	2.199	2450	2008	152	70	0	0	1	0
0.0470	−0.0177	−0.0179	0.0001	−0.0001	−0.0003	0.0100	26.377	0.000	0.000	0.000	0.000	0.001	0.4715	0.4655	0.4777	4721	7	2.817	2323	0	9.44e−05	0	1	198.1	2.206	2360	2105	175	79	0	1	1	0
0.0480	−0.0182	−0.0183	0.0002	−0.0001	−0.0003	0.0108	26.504	0.000	0.000	0.000	0.000	0.001	0.4724	0.4657	0.4792	4748	7	2.816	2325	0	8.70e−05	0	1	197.9	2.211	2291	2153	208	92	0	3	1	0
0.0490	−0.0186	−0.0188	0.0002	−0.0001	−0.0003	0.0117	26.627	0.000	0.000	0.000	0.000	0.001	0.4732	0.4655	0.4811	4773	7	2.815	2332	0	9.77e−05	0	1	197.7	2.217	2211	2219	233	105	0	4	1	0
0.0500	−0.0191	−0.0193	0.0002	−0.0001	−0.0003	0.0125	26.747	0.000	0.000	0.000	0.000	0.001	0.4740	0.4655	0.4825	4801	7	2.814	2338	0	8.37e−05	0	1	197.6	2.222	2168	2250	254	124	0	4	1	0
0.0510	−0.0196	−0.0198	0.0002	−0.0001	−0.0004	0.0134	26.863	0.000	0.000	0.000	0.000	0.001	0.4748	0.4663	0.4833	4828	7	2.814	2339	0	1.14e−04	0	1	197.4	2.228	2101	2304	276	136	2	8	1	0
0.0520	−0.0200	−0.0203	0.0002	−0.0001	−0.0004	0.0142	26.977	0.000	0.000	0.000	0.000	0.001	0.4754	0.4667	0.4842	4847	7	2.813	2341	0	9.57e−05	0	1	197.2	2.232	2038	2355	302	137	3	11	1	0
0.0530	−0.0205	−0.0208	0.0002	−0.0001	−0.0004	0.0151	27.087	0.000	0.000	0.000	0.000	0.001	0.4760	0.4675	0.4846	4863	7	2.812	2341	0	1.12e−04	0	1	197.0	2.237	1970	2396	327	151	4	14	1	0
0.0540	−0.0210	−0.0212	0.0002	−0.0001	−0.0004	0.0159	27.195	0.000	0.000	0.000	0.000	0.001	0.4766	0.4681	0.4852	4886	7	2.811	2339	0	1.09e−04	0	1	196.8	2.243	1922	2431	351	161	5	15	1	0
0.0550	−0.0214	−0.0217	0.0002	−0.0001	−0.0004	0.0168	27.300	0.000	0.000	0.000	0.000	0.001	0.4772	0.4692	0.4853	4903	8	2.810	2337	0	1.14e−04	0	1	196.7	2.247	1871	2460	386	160	7	17	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

This is the contents of the output file str_dir.dat. The two files str_nor.dat and str_tra.dat show the same data for the normal direction (2) and the transverse direction (3). Only the data for the 111-reflection are shown.

e_1	s_1	ep_P_1	ep_E_1	ep_E_2	ep_E_3	ep_E_4	ep_E_5	ep_E_6	g111	e111_1	s111_1
$0.00000e+00$	$0.00000e+00$	$0.00000e+00$	$0.00000e+00$	$0.00000e+00$	$0.00000e+00$	$0.00000e+00$	$0.00000e+00$	$0.00000e+00$	73	$0.00000e+00$	$0.00000e+00$
$2.50000e-02$	$1.77435e+01$	$5.59990e-19$	$2.50002e-04$	$-8.64076e-05$	$-8.67157e-05$	$4.74428e-08$	$-6.88762e-08$	$-6.47112e-07$	73	$2.42039e-04$	$1.82312e+01$
$2.60000e-02$	$1.84533e+01$	$5.85198e-19$	$2.60002e-04$	$-8.98637e-05$	$-9.01844e-05$	$4.93865e-08$	$-7.16977e-08$	$-6.73619e-07$	73	$2.51721e-04$	$1.89604e+01$
$2.70000e-02$	$1.91630e+01$	$6.11964e-19$	$2.70002e-04$	$-9.33199e-05$	$-9.36532e-05$	$5.13322e-08$	$-7.45223e-08$	$-7.00153e-07$	73	$2.61402e-04$	$1.96896e+01$
$2.80000e-02$	$1.98727e+01$	$6.36461e-19$	$2.80002e-04$	$-9.67760e-05$	$-9.71220e-05$	$5.32781e-08$	$-7.73470e-08$	$-7.26689e-07$	73	$2.71084e-04$	$2.04189e+01$
$2.90000e-02$	$2.05825e+01$	$6.59805e-19$	$2.90002e-04$	$-1.00232e-04$	$-1.00591e-04$	$5.52240e-08$	$-8.01717e-08$	$-7.53226e-07$	73	$2.80765e-04$	$2.11481e+01$
$3.00000e-02$	$2.12922e+01$	$6.87181e-19$	$3.00002e-04$	$-1.03688e-04$	$-1.04059e-04$	$5.71698e-08$	$-8.29964e-08$	$-7.79762e-07$	73	$2.90447e-04$	$2.18773e+01$
$3.10000e-02$	$2.20019e+01$	$7.13100e-19$	$3.10002e-04$	$-1.07144e-04$	$-1.07528e-04$	$5.91157e-08$	$-8.58212e-08$	$-8.06298e-07$	73	$3.00128e-04$	$2.26066e+01$
$3.20000e-02$	$2.27117e+01$	$7.37833e-19$	$3.20002e-04$	$-1.10601e-04$	$-1.10997e-04$	$6.10616e-08$	$-8.86459e-08$	$-8.32834e-07$	73	$3.09810e-04$	$2.33358e+01$
$3.30000e-02$	$2.33157e+01$	$1.52635e-04$	$3.28477e-04$	$-1.13527e-04$	$-1.13928e-04$	$6.38592e-08$	$-9.15738e-08$	$-8.56315e-07$	73	$3.18793e-04$	$2.39983e+01$
$3.40000e-02$	$2.37717e+01$	$5.18622e-04$	$3.34818e-04$	$-1.15709e-04$	$-1.16111e-04$	$6.64644e-08$	$-9.56289e-08$	$-8.71420e-07$	73	$3.26712e-04$	$2.45613e+01$
$3.50000e-02$	$2.41454e+01$	$1.00253e-03$	$3.39980e-04$	$-1.17480e-04$	$-1.17883e-04$	$6.88766e-08$	$-9.78960e-08$	$-8.82711e-07$	73	$3.33981e-04$	$2.50652e+01$
$3.60000e-02$	$2.44549e+01$	$1.57846e-03$	$3.44221e-04$	$-1.18932e-04$	$-1.19333e-04$	$7.14421e-08$	$-9.99214e-08$	$-8.90369e-07$	73	$3.40701e-04$	$2.55203e+01$
$3.70000e-02$	$2.47103e+01$	$2.23142e-03$	$3.47693e-04$	$-1.20116e-04$	$-1.20515e-04$	$7.31243e-08$	$-1.01245e-07$	$-8.94015e-07$	73	$3.46919e-04$	$2.59321e+01$
$3.80000e-02$	$2.49331e+01$	$2.93055e-03$	$3.50703e-04$	$-1.21138e-04$	$-1.21537e-04$	$7.64789e-08$	$-1.02637e-07$	$-8.96651e-07$	73	$3.52804e-04$	$2.63162e+01$
$3.90000e-02$	$2.51353e+01$	$3.65904e-03$	$3.53418e-04$	$-1.22058e-04$	$-1.22457e-04$	$8.01678e-08$	$-1.04910e-07$	$-8.98739e-07$	73	$3.58456e-04$	$2.66817e+01$
$4.00000e-02$	$2.53233e+01$	$4.40755e-03$	$3.55934e-04$	$-1.22909e-04$	$-1.23307e-04$	$8.40244e-08$	$-1.05889e-07$	$-9.01014e-07$	73	$3.63936e-04$	$2.70340e+01$
$4.10000e-02$	$2.54979e+01$	$5.17481e-03$	$3.58262e-04$	$-1.23694e-04$	$-1.24093e-04$	$8.79099e-08$	$-1.05984e-07$	$-9.01549e-07$	73	$3.69244e-04$	$2.73733e+01$
$4.20000e-02$	$2.56625e+01$	$5.95616e-03$	$3.60449e-04$	$-1.24432e-04$	$-1.24830e-04$	$9.15821e-08$	$-1.06056e-07$	$-9.02362e-07$	73	$3.74412e-04$	$2.77022e+01$
$4.30000e-02$	$2.58185e+01$	$6.74952e-03$	$3.62516e-04$	$-1.25127e-04$	$-1.25526e-04$	$9.49428e-08$	$-1.05670e-07$	$-9.02478e-07$	73	$3.79456e-04$	$2.80220e+01$
$4.40000e-02$	$2.59670e+01$	$7.55336e-03$	$3.64478e-04$	$-1.25786e-04$	$-1.26185e-04$	$9.78245e-08$	$-1.06024e-07$	$-9.02053e-07$	73	$3.84385e-04$	$2.83335e+01$
$4.50000e-02$	$2.61089e+01$	$8.36645e-03$	$3.66347e-04$	$-1.26443e-04$	$-1.26813e-04$	$1.00523e-07$	$-1.06618e-07$	$-9.00971e-07$	73	$3.89210e-04$	$2.86375e+01$
$4.60000e-02$	$2.62453e+01$	$9.18716e-03$	$3.68141e-04$	$-1.27015e-04$	$-1.27414e-04$	$1.02740e-07$	$-1.07153e-07$	$-9.00372e-07$	73	$3.93942e-04$	$2.89349e+01$
$4.70000e-02$	$2.63766e+01$	$1.00148e-02$	$3.69864e-04$	$-1.27592e-04$	$-1.27991e-04$	$1.05275e-07$	$-1.07461e-07$	$-9.00098e-07$	73	$3.98586e-04$	$2.92262e+01$
$4.80000e-02$	$2.65038e+01$	$1.08483e-02$	$3.71530e-04$	$-1.28150e-04$	$-1.28548e-04$	$1.07686e-07$	$-1.07071e-07$	$-8.99731e-07$	73	$4.03156e-04$	$2.95124e+01$
$4.90000e-02$	$2.66271e+01$	$1.16870e-02$	$3.73143e-04$	$-1.28690e-04$	$-1.29087e-04$	$1.10395e-07$	$-1.06823e-07$	$-8.99290e-07$	73	$4.07653e-04$	$2.97936e+01$
$5.00000e-02$	$2.67470e+01$	$1.25304e-02$	$3.74709e-04$	$-1.29214e-04$	$-1.29609e-04$	$1.12771e-07$	$-1.06336e-07$	$-8.99015e-07$	73	$4.12089e-04$	$3.00704e+01$
$5.10000e-02$	$2.68633e+01$	$1.33788e-02$	$3.76226e-04$	$-1.29721e-04$	$-1.30115e-04$	$1.14904e-07$	$-1.06250e-07$	$-8.98707e-07$	73	$4.16453e-04$	$3.03423e+01$
$5.20000e-02$	$2.69767e+01$	$1.42311e-02$	$3.77703e-04$	$-1.30215e-04$	$-1.30607e-04$	$1.17063e-07$	$-1.06339e-07$	$-8.97952e-07$	73	$4.20759e-04$	$3.06103e+01$
$5.30000e-02$	$2.70873e+01$	$1.50874e-02$	$3.79140e-04$	$-1.30695e-04$	$-1.31085e-04$	$1.19116e-07$	$-1.06377e-07$	$-8.97075e-07$	73	$4.25003e-04$	$3.08742e+01$
$5.40000e-02$	$2.71949e+01$	$1.59477e-02$	$3.80538e-04$	$-1.31162e-04$	$-1.31550e-04$	$1.20785e-07$	$-1.06098e-07$	$-8.96193e-07$	73	$4.29188e-04$	$3.11340e+01$
$5.50000e-02$	$2.73001e+01$	$1.68114e-02$	$3.81902e-04$	$-1.31616e-04$	$-1.32003e-04$	$1.22667e-07$	$-1.05574e-07$	$-8.95408e-07$	73	$4.33318e-04$	$3.13901e+01$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

This is the contents of the output file stddev1.dat. The two files stddev2.dat and stddev3.dat show the same data for the normal direction (2) and the transverse direction (3). Only the data for the first four reflections are shown.

e_1	s_1	ep_P_1	$e111$	$s111$	$e200$	$s200$	$e220$	$s220$	$e311$	$s311$
$0.00000e+00$	$0.00000e+00$	$0.00000e+00$	$8.56645e-08$	$5.24344e-03$	$1.19424e-07$	$7.30880e-03$	$1.51027e-07$	$9.23784e-03$	$1.52471e-06$	$9.33715e-02$
$2.50000e-02$	$1.77435e+01$	$5.59990e-19$	$8.56645e-08$	$5.24344e-03$	$1.19424e-07$	$7.30880e-03$	$1.51027e-07$	$9.23784e-03$	$1.52471e-06$	$9.33715e-02$
$2.60000e-02$	$1.84533e+01$	$5.85198e-19$	$8.90914e-08$	$5.45319e-03$	$1.24202e-07$	$7.60118e-03$	$1.57070e-07$	$9.60743e-03$	$1.58569e-06$	$9.71063e-02$
$2.70000e-02$	$1.91630e+01$	$6.11964e-19$	$9.25183e-08$	$5.66294e-03$	$1.28979e-07$	$7.89357e-03$	$1.63113e-07$	$9.97703e-03$	$1.64668e-06$	$1.00841e-01$
$2.80000e-02$	$1.98727e+01$	$6.36461e-19$	$9.59453e-08$	$5.87270e-03$	$1.33757e-07$	$8.18595e-03$	$1.69155e-07$	$1.03466e-02$	$1.70767e-06$	$1.04576e-01$
$2.90000e-02$	$2.05825e+01$	$6.59805e-19$	$9.93722e-08$	$6.08245e-03$	$1.38535e-07$	$8.47834e-03$	$1.75198e-07$	$1.07162e-02$	$1.76865e-06$	$1.08311e-01$
$3.00000e-02$	$2.12922e+01$	$6.87181e-19$	$1.02799e-07$	$6.29220e-03$	$1.43312e-07$	$8.77072e-03$	$1.81241e-07$	$1.10858e-02$	$1.82964e-06$	$1.12045e-01$
$3.10000e-02$	$2.20019e+01$	$7.13100e-19$	$1.06226e-07$	$6.50196e-03$	$1.48090e-07$	$9.06310e-03$	$1.87284e-07$	$1.14554e-02$	$1.89063e-06$	$1.15780e-01$
$3.20000e-02$	$2.27117e+01$	$7.37833e-19$	$1.09653e-07$	$6.71171e-03$	$1.52867e-07$	$9.35549e-03$	$1.93326e-07$	$1.18250e-02$	$1.95161e-06$	$1.19515e-01$
$3.30000e-02$	$2.33157e+01$	$1.52635e-04$	$1.12965e-07$	$6.90173e-03$	$1.57497e-07$	$9.62088e-03$	$1.99288e-07$	$1.21651e-02$	$2.01057e-06$	$1.22899e-01$
$3.40000e-02$	$2.37717e+01$	$5.18622e-04$	$1.16103e-07$	$7.06316e-03$	$1.61880e-07$	$9.84556e-03$	$2.04921e-07$	$1.24426e-02$	$2.06623e-06$	$1.25763e-01$
$3.50000e-02$	$2.41454e+01$	$1.00253e-03$	$1.19121e-07$	$7.20709e-03$	$1.66104e-07$	$1.00461e-02$	$2.10486e-07$	$1.26898e-02$	$2.83223e-06$	$1.67062e-01$
$3.60000e-02$	$2.44549e+01$	$1.57846e-03$	$1.22029e-07$	$7.33637e-03$	$1.70176e-07$	$1.02260e-02$	$2.15939e-07$	$1.29067e-02$	$3.81433e-06$	$2.34821e-01$
$3.70000e-02$	$2.47103e+01$	$2.23142e-03$	$1.24831e-07$	$7.45263e-03$	$1.74104e-07$	$1.03872e-02$	$1.33351e-06$	$8.09505e-02$	$4.18483e-06$	$2.66723e-01$
$3.80000e-02$	$2.49331e+01$	$2.93055e-03$	$1.27570e-07$	$7.56037e-03$	$1.77947e-07$	$1.05365e-02$	$2.62371e-06$	$1.43714e-01$	$4.11338e-06$	$2.71664e-01$
$3.90000e-02$	$2.51353e+01$	$3.65904e-03$	$1.30272e-07$	$7.66229e-03$	$1.81740e-07$	$1.06773e-02$	$3.33393e-06$	$1.78044e-01$	$3.98924e-06$	$2.76313e-01$
$4.00000e-02$	$2.53233e+01$	$4.40755e-03$	$1.32938e-07$	$7.75966e-03$	$1.74427e-06$	$6.67870e-02$	$3.79990e-06$	$2.00529e-01$	$3.84459e-06$	$2.79206e-01$
$4.10000e-02$	$2.54979e+01$	$5.17481e-03$	$1.35572e-07$	$7.85255e-03$	$3.11222e-06$	$1.24896e-01$	$3.83897e-06$	$2.02331e-01$	$3.66039e-06$	$2.81883e-01$
$4.20000e-02$	$2.56625e+01$	$5.95616e-03$	$1.38184e-07$	$7.94174e-03$	$3.97330e-06$	$1.63119e-01$	$3.84563e-06$	$2.02398e-01$	$3.47577e-06$	$2.84882e-01$
$4.30000e-02$	$2.58185e+01$	$6.74952e-03$	$1.40779e-07$	$8.02780e-03$	$4.28998e-06$	$1.77076e-01$	$3.84763e-06$	$2.02184e-01$	$3.27704e-06$	$2.87637e-01$
$4.40000e-02$	$2.59670e+01$	$7.55336e-03$	$1.43351e-07$	$8.11081e-03$	$4.38631e-06$	$1.81460e-01$	$3.82267e-06$	$2.00633e-01$	$3.07289e-06$	$2.90162e-01$
$4.50000e-02$	$2.61089e+01$	$8.36645e-03$	$1.45905e-07$	$8.19114e-03$	$4.22100e-06$	$1.74143e-01$	$3.78414e-06$	$1.98405e-01$	$2.87050e-06$	$2.93040e-01$
$4.60000e-02$	$2.62453e+01$	$9.18716e-03$	$1.48442e-07$	$8.26909e-03$	$4.04871e-06$	$1.66721e-01$	$3.75875e-06$	$1.96847e-01$	$2.66750e-06$	$2.95467e-01$
$4.70000e-02$	$2.63766e+01$	$1.00148e-02$	$1.50956e-07$	$8.34465e-03$	$3.91502e-06$	$1.61227e-01$	$3.71398e-06$	$1.94384e-01$	$2.47482e-06$	$2.97672e-01$
$4.80000e-02$	$2.65038e+01$	$1.08483e-02$	$1.53450e-07$	$8.41823e-03$	$3.77543e-06$	$1.55493e-01$	$3.66797e-06$	$1.91880e-01$	$2.29684e-06$	$3.00129e-01$
$4.90000e-02$	$2.66271e+01$	$1.16870e-02$	$1.55929e-07$	$8.48987e-03$	$3.69985e-06$	$1.52665e-01$	$3.61171e-06$	$1.88894e-01$	$2.13981e-06$	$3.03266e-01$
$5.00000e-02$	$2.67470e+01$	$1.25304e-02$	$1.58392e-07$	$8.55994e-03$	$3.66647e-06$	$1.51714e-01$	$3.56247e-06$	$1.86277e-01$	$2.01901e-06$	$3.07504e-01$
$5.10000e-02$	$2.68633e+01$	$1.33788e-02$	$1.60843e-07$	$8.62827e-03$	$3.65310e-06$	$1.51209e-01$	$3.51609e-06$	$1.83730e-01$	$1.91153e-06$	$3.11847e-01$
$5.20000e-02$	$2.69767e+01$	$1.42311e-02$	$1.63280e-07$	$8.69510e-03$	$3.63479e-06$	$1.50444e-01$	$3.46954e-06$	$1.81316e-01$	$1.82634e-06$	$3.15930e-01$
$5.30000e-02$	$2.70873e+01$	$1.50874e-02$	$1.65704e-07$	$8.76032e-03$	$3.61808e-06$	$1.49733e-01$	$3.42000e-06$	$1.78779e-01$	$1.76596e-06$	$3.20199e-01$
$5.40000e-02$	$2.71949e+01$	$1.59477e-02$	$1.68111e-07$	$8.82399e-03$	$3.60557e-06$	$1.49217e-01$	$3.36795e-06$	$1.76128e-01$	$1.72126e-06$	$3.24599e-01$
$5.50000e-02$	$2.73001e+01$	$1.68114e-02$	$1.70501e-07$	$8.88630e-03$	$3.57619e-06$	$1.47953e-01$	$3.32003e-06$	$1.73733e-01$	$1.70505e-06$	$3.28996e-01$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

References

Bunge, H.-J. (1982). *Texture Analysis in Materials Science.* Butterworths & Co.

Clausen, B. (1997). Characterisation of Polycrystal Deformation by Numerical Modelling and Neutron Diffraction Experiments. Thesis, *Risø–R–985*, Risø National Laboratory, 4000 Roskilde, Denmark.

Dieter, G.E. (1988). *Mechanical Metallurgy.* McGraw-Hill.

Eshelby, J.D. (1957). The Determination of the Elastic Field of an Ellipsoidal Inclusion, and Related Problems. *Proc. Roy. Soc. Lond.* **A241**, 376.

Hill, R. (1965a). Continuum Micro-mechanics of Elastoplastic Polycrystals. *J. Mech. Phys. Solids* **13**, 89.

Hill, R. (1965b). A Self-consistent Mechanics of Composite Materials. *J. Mech. Phys. Solids* **13**, 231.

Hill, R. (1966). Generalized Constitutive Relations for Incremental Deformation of Metal Crystals by Multislip. *J. Mech. Phys. Solids* **14**, 95.

Hutchinson, J. W. (1970). Elastic-plastic Behaviour of Polycrystalline Metals and Composites. *Proc. Roy. Soc. Lond.* **A319**, 247.

Kneer, G. (1965). Berechnung der Elastischen Konstanten des Vielkristalls aus den Konstanten des Einkristalls. *Phys. Stat. Sol.* **9**, 825.

Ledbetter, H.M. (1984). Monocrystal-Polycrystal Elastic Constants of a Stainless Steel. *Phys. Stat. Sol.* **A 85**, 89.

Pedersen, P. (1995). Simple transformations by proper contracted forms: Can we change the usual practice? *Numerical Methods in Engineering* **11**, 821.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992). *Numerical Recipes in C*, Cambridge University Press 430-444.

Sachs, G. (1928). Zur Ableitung einer Fließbedingung. *Z. Ver. Deu. Ing.* **72-22**, 734.

Taylor, G.I. (1938). Plastic Strain in Metals. *J. Inst. Metals* **62** 307.

Title and author(s)

A Self-consistent Model for Polycrystal Deformation
Description and Implementation

Bjørn Clausen and Torben Lorentzen

Abstract (Max. 2000 char.)

This report is a manual for the ANSI C implementation of an incremental elastic-plastic rate-insensitive self-consistent polycrystal deformation model based on (Hutchinson 1970). The model is furthermore described in the Ph.D. thesis by Clausen (Clausen 1997). The structure of the main program, `sc_model.c`, and its subroutines are described with flow-charts. Likewise the pre-processor, `sc_ini.c`, is described with a flowchart. Default values of all the input parameters are given in the pre-processor, but the user is able to select from other pre-defined values or enter new values. A sample calculation is made and the results are presented as plots and examples of the output files are shown.

Descriptors INIS/EDB